A CYBER INFRASTRUCTURE FOR HARD AND SOFT DATA FUSION

A Thesis in
Electrical Engineering

by

Kaustubh Misra

© 2010 Kaustubh Misra

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2010
The thesis of Kaustubh Misra was reviewed and approved* by the following:

David Hall  
Interim Dean,  
Professor of Information Sciences and Technology  
Thesis Advisor

John F Doherty  
Professor of Electrical Engineering

Kenneth Jenkins  
Professor and Head of Department  
Electrical Engineering

*Signatures are on file in the Graduate School
ABSTRACT

Multi-sensor Data Fusion; utilizing sensors of various natures and denominations, is gradually finding implementation in many fields from Defense and Security to “Human Landscape” Evaluation and Environment Impact Assessment. It is an imperative tool in analyzing information about a complex situation or event and coming to a concrete conclusion about it utilizing sensing resources at hand. Apart from using the traditional sensors to assimilate information about targets, the focus is now shifting towards including a vital piece of information; observing and understanding what is called the “Human Terrain” and Human Landscape or the complex interrelationships and trends that define a specific human population. This thesis takes traditional sensing and furthers it with the addition of Human Observers. In consonance with using humans or “Soft Sensors”, the mapping of users of Microblogs as sensor arrays to facilitate Information extraction is analyzed. A cyber infrastructure is described to help realize the idea of fusing information from Human Observers, Traditional Sensors and web based entities. Also, a Distributed Multiple Software Agent System has been implemented to demonstrate the idea of Dynamic Human Sensor Array creation over the web. An infrastructure has been developed and explained that employs traditional sensors like Closed Circuit Video Camera Systems in conjunction with this Multiple Agent System to build a Multisensor Data Fusion Environment. Finally, an Ad-Hoc Wireless Sensor Network Creation Algorithm is described that uses Bluetooth. This adds a physical sensor array development example to the cyber infrastructure.
# TABLE OF CONTENTS

**LIST OF FIGURES** ................................................................................................................. vi

**LIST OF TABLES** ................................................................................................................... ix

**ACKNOWLEDGEMENTS** ..................................................................................................... x

Chapter 1  Introduction ............................................................................................................ 1
  - Evolution of Information Fusion ...................................................................................... 2
  - Emergence of New Information Sources ......................................................................... 3
    - Soft Sensing .................................................................................................................. 3
    - S-Space ......................................................................................................................... 4
    - H-Space ......................................................................................................................... 4
    - I-Space ......................................................................................................................... 5
  - Applications to characterizing the Human Landscape ..................................................... 5
  - Fusion of emerging data sources ...................................................................................... 8

Chapter 2  Related Work .......................................................................................................... 10

Chapter 3  Background Study .................................................................................................. 12
  - The JDL Data Fusion Model ............................................................................................ 12
    - Level 0 Fusion ............................................................................................................. 13
    - Level 1 Fusion ............................................................................................................. 14
    - Level 2 Fusion ............................................................................................................. 14
    - Level 3 Fusion ........................................................................................................... 18
    - Level 4 Fusion ........................................................................................................... 20
    - Level 5 Processing ................................................................................................... 20
  - Participatory Sensing ..................................................................................................... 21
  - Multi Agent Systems ..................................................................................................... 26
  - I-Space Introduction: Internet as a Data Source ............................................................... 30

Chapter 4 Architectural Requirements for enabling hard and soft data fusion ..................... 34
  - Functional Architecture .................................................................................................. 34
    - Soft Sensor Characterization ....................................................................................... 37
    - User Generated Content: I-Space ............................................................................... 39
    - Human Observation Reports: H-Space ....................................................................... 42

Chapter 5 A Prototype Cyber Infrastructure Environment: I-Space .................................... 44
  - Design and Implementation ............................................................................................ 44
    - Principle ....................................................................................................................... 45
    - General Design Decisions ......................................................................................... 45
    - Sensor Array Characterization Layer Design .............................................................. 49
    - Cougaar Software Agent Society Structure .............................................................. 50
| Chapter 6 A Prototype Cyber Infrastructure Environment: S-Space .......................................................... 71 |
|---------------------------------------------------------------|-------------------|
| Design and Implementation .................................................. 71 |
| Physical and Network Infrastructure ...................................... 72 |
| Object Tracking Algorithms ................................................... 73 |
| Dataset Details ........................................................................ 73 |
| Covariance Tracking ................................................................. 74 |
| Image Segmentation using Frame Differencing and Motion History ........................................... 80 |
| The Histogram of Oriented Gradients (HOG) Algorithm .................. 87 |

| Chapter 7 Wireless Ad-Hoc Mesh Network Formation over Bluetooth .............................................. 96 |
|-------------------------------------------------------------------------------------------------|-------------------|
| Bluetooth Basics ....................................................................................................................... 96 |
| Bluetooth Operation Concepts ................................................................................................... 98 |
| National Semiconductor LMX9838 Bluetooth Modules ................................................................. 102 |
| Bluetooth Sensor Network Creation Algorithm ........................................................................... 103 |
| Bluetooth Ad-Hoc Piconet Development Procedure .................................................................... 104 |
| Routing Protocol Design Details ............................................................................................... 109 |
| Conclusion .................................................................................................................................... 112 |

| Chapter 8 Application Scenarios .................................................................................................. 113 |

| Chapter 9 Summary and Recommendations ................................................................................... 117 |
|-------------------------------------------------------------------------------------------|-------------------|
| Limitations and Future Research ............................................................................................ 117 |
| Experiment Design Requirements ............................................................................................... 118 |
| Result Analysis and Conclusion ............................................................................................... 118 |

| References .................................................................................................................................. 120 |

| Appendix A: Twitter Terminology ............................................................................................... 128 |

| Appendix B: Database Schema .................................................................................................... 130 |
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>JDL Data Fusion Model</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Level 2 Data Fusion Schematic</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Bayesian Belief Network</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Level 3 Data Fusion Schematic</td>
<td>19</td>
</tr>
<tr>
<td>3.5</td>
<td>Participatory Sensing Schematic</td>
<td>22</td>
</tr>
<tr>
<td>3.6</td>
<td>Intelligent Agent and its Intrinsic Functionality</td>
<td>27</td>
</tr>
<tr>
<td>3.7</td>
<td>Web Based Microblogs as Sensor Arrays</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Cyber Data Fusion Architecture</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>I Space Applications Interface</td>
<td>46</td>
</tr>
<tr>
<td>5.2</td>
<td>Dynamic Soft Sensor Array Creation Architecture</td>
<td>48</td>
</tr>
<tr>
<td>5.3</td>
<td>Cougaar Society Structure</td>
<td>50</td>
</tr>
<tr>
<td>5.4</td>
<td>Dynamic Array Worldwide Soft Sensor Distribution</td>
<td>56</td>
</tr>
<tr>
<td>5.5</td>
<td>Dynamic Array USA State Wide Soft Sensor Distribution</td>
<td>57</td>
</tr>
<tr>
<td>5.6</td>
<td>Dynamic Array USA City Wide Soft Sensor Distribution</td>
<td>58</td>
</tr>
<tr>
<td>5.7</td>
<td>Dynamic Array Time Zone Based Soft Sensor Distribution</td>
<td>59</td>
</tr>
<tr>
<td>5.8</td>
<td>Message Exchange Architecture</td>
<td>60</td>
</tr>
<tr>
<td>5.9</td>
<td>XML Based Servlet Response Format for Contact Lists</td>
<td>62</td>
</tr>
<tr>
<td>5.10</td>
<td>Servlet Response for Contact List Request</td>
<td>62</td>
</tr>
<tr>
<td>5.11</td>
<td>XML Based Servlet Response Format for Contact Information</td>
<td>63</td>
</tr>
<tr>
<td>5.12</td>
<td>Servlet Response for Contact Information Request</td>
<td>64</td>
</tr>
<tr>
<td>5.13</td>
<td>Higher Layer Design Considerations</td>
<td>65</td>
</tr>
<tr>
<td>6.1</td>
<td>Video Processing Infrastructure</td>
<td>72</td>
</tr>
</tbody>
</table>
Figure 6.2: Single Non Rigid Object Tracking using Covariance Matrices (ETH) ..... 77
Figure 6.3: Single Non Rigid Object Covariance Tracking (IST) ..................... 78
Figure 6.4: Multiple Non Rigid Object Covariance Tracking (ETH) ............... 79
Figure 6.5: Movement Vector Plot for 2 Objects, Covariance Tracking (ETH) .. 80
Figure 6.6: Frame Diff. with Decay Factor 250 and Threshold 80 (IST) ............ 83
Figure 6.7: Frame Diff. with Decay Factor 50 and Threshold 80 (IST) ............. 84
Figure 6.8: Frame Diff. with Decay Factor 250 and Threshold 20 (IST) ............ 85
Figure 6.9: Frame Diff. Detected Sequence Decay 250, Threshold 80 (IST) ...... 85
Figure 6.10: Frame Diff. Multiple Object Detection Decay 50, Threshold 8 (IST) .. 86
Figure 6.11: Frame Diff. Multiple Object Detection Weaknesses (IST) .......... 87
Figure 6.12: Histogram of Oriented Gradients Person Test Images .................. 89
Figure 6.13: Histogram of Oriented Gradients Non-Person Test Images ........... 90
Figure 6.14: Histogram of Oriented Gradients Non-Person image a Result ...... 90
Figure 6.15: Histogram of Oriented Gradients Non-Person image b Result ...... 91
Figure 6.16: Histogram of Oriented Gradients Non-Person image c Result ...... 91
Figure 6.17: Histogram of Oriented Gradients Non-Person image d Result ...... 92
Figure 6.18: Histogram of Oriented Gradients Person image a Result ............ 92
Figure 6.19: Histogram of Oriented Gradients Person image b Result ............ 93
Figure 6.20: Histogram of Oriented Gradients Person image c Result ............ 93
Figure 6.21: Histogram of Oriented Gradients Person image d Result ............ 94
Figure 6.22: Histogram of Oriented Gradients Person image e Result ............ 94
Figure 7.1: The Bluetooth Protocol Stack ...................................................... 99
Figure 7.2: A Bluetooth Piconet using the SPP Bluetooth Profile ................. 104
Figure 7.3: Host – LMX9838 Connection Schematic .............................................. 105
Figure 7.4: LMX9838 SPP Packet Format ............................................................. 109
Figure 7.5: Payload Format for Data Link Layer Routing ...................................... 111
Figure 8.1: Extreme Event Prediction: General Idea ............................................ 113
Figure 8.2: Agent Hierarchy .................................................................................. 116
LIST OF TABLES

Table 6.1: Color Code for Motion History Movement ......................................................... 82

Table 7.1: Bluetooth Class Based Range .............................................................................. 98
ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my advisor Dr. David Hall, for his valuable suggestions, guidance and constant encouragement. Dr Hall has played an instrumental role in my research and this thesis would not have been a reality without his support. I also thank Dr John Doherty for his patience and support. I also thank Brian Panulla, Jeffery Rimland and Hemant Kumar at the Center for Network Centric Cognition and Information Fusion for their insights and suggestions. Finally, I also thank my family who encouraged me throughout my Masters at Penn State.
Chapter 1

Introduction

The study of complex human interactions in space within the ecosystem has given birth to a field broadly known as Data Fusion. Data Fusion refers to the act of deriving meaningful information from an original raw dataset that might be the output of a sensor. Multi-sensor Data Fusion [1], as it is contemporarily termed, is a vital tool in analyzing a complex situation and coming to a concrete conclusion about it. Apart from using the traditional “Hard” sensors like physical sensors to assimilate information about physical targets, the focus is now shifting towards including a vital piece of information; i.e. observing and understanding what is called the “Human Terrain”. The term Human Terrain represents complex interrelationships and trends that define a specific human population. This information adds a new dimension to the already well researched physical sensor domain. In addition to this, the new concept of using humans as “Soft Sensors” is being introduced that utilizes human senses and the inherent capabilities of humans to understand their surroundings and complex relationships therein.

The traditional view of data fusion that utilized physical hard sensors to isolate specific information about physical targets is being changed due to the changes in our understanding of the nature of the targets themselves. The targets no longer remain physical entities in the sense that vital information is to be derived from the understanding of their origin or nature. The interest in finding not only location but identity and the relationship of targets with their surroundings makes this a more complex
The interactions of any target with its surroundings, whether hostile or not, are important in coming to any conclusion about its actual nature. Further, an understanding of non-physical attributes like human networks, intent, policies, beliefs etc is important to understand the physical target that projects them. This makes it important to study and understand the “Human Landscape” apart from the traditional “Physical Landscape” as sensed by traditional sensing methodologies.

**Evolution of Information Fusion**

The discipline of information fusion has been evolving at an astonishing pace after the interest shown into it by the USA Department of Defense (DoD). From the initial stages of the conception of the DoD Joint Directors of Laboratories (JDL) Data Fusion Model, various new ideas and concepts have been introduced. [2] gives a good overview about the DoD funded data fusion systems that existed as of early 1991. From the mid 90’s onwards, with the steady increase in computational capabilities, and the emergence of image processing, various new sensors like video cameras can now be used to extract information from, using software principally without the need for human involvement.

The Handbook of Multisensor Data fusion [3] also gives a good overview into the techniques that have been generally employed towards creating the best fusion practices. However, in the contemporary world, where traditionally existing sensor infrastructures are increasingly being challenged with understanding Human behavior, it has been realized that humans themselves might be instrumental in doing so. Hence the traditional
view of sensors and fusion has evolved to the stage that human beings are no longer considered to be just monitoring entities. They are increasingly being considered sensors themselves, with a lot to contribute since they best understand their immediate surroundings.

**Emergence of New Information Sources**

**Soft Sensing**

Progress in the sphere of web-based human internetworking infrastructure and the growing human participation in reporting events in their immediate surroundings have opened new avenues when it comes to sensor characterization. Two new sources of sensorial data have emerged, namely, Human (Soft) Sensors and Web Based Information Sensors. The ease with which humans can now share information has ushered in a wave of new age ‘public reporters’ that form a big source of immediate information about the world that surrounds them. Apart from humans reporting about events directly, the web also forms a big platform for information sharing and distribution and coupled together with the new human instinct to report, it needs to be mapped and fully understood to complement existing data fusion systems. These new avenues form what is called “Soft Sensing”. In other words then, the Traditional Sensing Domain is now joined by the Human Sensor Domain as well as the Internet or Web Domain.

While, there are numerous challenges in utilizing these data sources such as the inherent uncertainty of human behavior, deliberate falsification of data, pure error in
judgment, data corruption, data collection, data characterization, human modeling, data source identity authentication, privacy, motivation and second order effects such as rumor effects; this still remains a huge dataset of usable information and this thesis will address some of the issues using the proposed model architecture.

**S-Space**

This report uses the term S-Space to designate all the types of data sources which can be described as traditional sensing resources or hard sensors. An S-Space sensor is a man made device that receives and responds to an input signal and converts it into its equivalent electrical form. Examples include active and passive sensors like temperature sensors, pressure sensors, Global Positioning System Devices, Video Cameras, Audio Microphones, RADARS, SONARS, LIDARS and so on. So S-Space Sensors measure physical quantities and convert them into a Human Observer or Instrument readable format.

**H-Space**

The term “H-Space” represents Dynamic communities of Human Observers. Human Observers might be further categorized on the basis of their formal training which accounts for their accuracy to perceive the input signal. So an Array of Humans reporting events via different means lie in the H-Space.
I-Space

The term “I-Space” represents the World Wide Web or the Internet Space comprising of web data from a myriad of sources like social networks, blogs, reports, news and archived sensor data. In a sense, H-Space overlaps I-Space in cases where Human Observers directly transmit information about their surroundings via web interfaces such as blogs. However, the I-Space is in fact a unique channel of data transmission in that it most definitely involves transmission of S-Space data as well. The H-Space and I-Space combine to form what was previously categorized as soft sensing.

Applications to Characterizing the Human Landscape

The Human Landscape can be defined as “people, in the aggregate at various scales, as they relate to the natural world, the built environment, and each other,” [4]. The Human Terrain was a term coined by the United States Army to represent local cultures. This concept helps relate the physical environment of human beings (the traditional landscape as physically modified by humans) to actual human behavior in different contexts. Some contexts that bear importance in today’s world include business planning, emergency crisis management, security systems and surveillance for national and international security, environmental monitoring and impact assessment.

There are several concepts related to the general idea of characterizing the human landscape. For example, the human landscape is characterized at the aggregate level and not individual level [4]. So people are viewed as “groups”. People can be formed into distinct groups by identifying key geographic, sociological, economic, political, and even
environmental features. These groups of people have distinct relationships with their natural physical surroundings. Not only do they depend on this natural environment, they also depend on the changes that they bring about to them. For example, a city which is a man made physical landscape is directly representative of the group of people it belongs to. Their specific traits also reflect in the design of this city. Besides this interaction, there are other complex interactions between groups of people. Hence, Characterizing the human landscape is a multifaceted, multidimensional problem due to these various interactions between the living and non-living entities.

According to [4] the major issues in characterizing the Human Landscape include the presence of tacit knowledge along with explicit knowledge as far has human dealings are concerned, the dependence of Human response on various factors such as relationships with other human beings surrounding them as well as the importance of context in determining the ultimate nature of any relationship which includes the characterization of the source of information. Apart from these, there are the other issues of representing uncertainty, representing time along with Human behavior models, isolating the factors that affect human judgment whether knowingly or unknowingly of the Human Sensor like incentives and risk.

Among all of these the primary question arises, in order for us to most accurately characterize the Human Landscape, is it not absolutely necessary to most accurately characterize human sensors? Absolutely; and in doing that, the following questions need to be answered:
1. What is the Human Sensor trying to convey? This report benefits who? What incentive would the human have in reporting this event? This analysis will initialize the system’s understanding of the nature and circumstances of the report.

2. Who is the Human Sensor Reporting the data: What is the nature of the sensor? What is the source of their information? If they are not the origin what is and how accurate is that? If they indeed are the origin, how trustworthy are they given the context and situation. So along with the what, the how also needs to be answered here.

3. When and where was the situation encountered? Are there any other statistically relevant sources of similar information for the same region? Area dependent phenomena can be for example monitored easily if it is known that a statistical majority report it from the given region with a certain confidence interval.

4. The final question is which elements is the report pointing to? What effect do those elements have on the Local Human Landscape? Is that detrimental or beneficial to the observed entity within context?

Traditionally, with S-Space Information Fusion for characterizing the physical landscape, physical attributes of the environment like for example climatic factors and their corresponding scalar or vector quantity arrays can be modeled to a state vector that represents the physical landscape through what is known as an observation equation. In this case though, with soft sensors (Humans) in the field this situation cannot be modeled directly but in fact the inverse can be better visualized. Hence, we begin with an estimated model state vector that would most closely characterize any target entity. With this model, predictions can be made via the observation equation on the future of the
evolving situation within context of the target. Observational error limits and margins can also be studied to provide the desired level of reliability into the model. A feedback loop will also ensure that this observational error is compensated for through subsequent iterations of the model development process.

Hence, with Human sensors, fusion can be achieved by adding redundancy into the system. What this essentially means is that whatever individual sensor feeds lack in accuracy will be made up for by the sheer number of independent and noisy sources of information. With these observations, finding the state vector that best represents the observations then becomes a pure estimation problem. This has been detailed in texts such as [5].

As stated in [4], an emerging representation of a Human Sensor array is to focus on network representations i.e. to characterize individuals as nodes and links between individuals as relationships between these nodes. Then, sub-network matching techniques and network dynamic tools can be used. However, this is by no means an all-inclusive representation or estimation method, despite its mathematical elegance. Extensive work must be done to define quantities such as “observables”, “state vectors”, “observation equations”, and “equations of motion” concepts that apply to the human landscape.

**Fusion of emerging data sources**

Now that the importance of Humans working as sensors and the importance of the subsequent help in characterizing the human landscape are clear, the process must begin with the identification of the scope of information retrieval. What this means is that it
must be understood before beginning that what type of information can and cannot be collected in a given scenario and geographical context. Next is the formation of the resource pool or identifying key sensors and creation of a Sensor Array that is best suited to the context at hand. Closely monitoring the situation to be assessed and the resources needed is crucial to the success of a particular effort. For example, a terrorist threat detection system would need skilled event reporters and soldiers as a major part of the human sensor array, whereas for an environmental detection system, the general public with their vast geographical footprint would be suited to cover a wider area. Besides, it would tolerate way more noise that the former case where report accuracy becomes crucial.

Observing large quantities of noisy data of variable reliability, channeling information into individually manageable streams and getting meaning out of each individual stream will form the basis of the system used for human landscape information fusion. Also, conveying this information to the most relevant user groups is also critical to the systems overall success.
Chapter 2

Related Work

Many efforts to develop a system to combine information from various sensors have been made around the globe. [6] Presents an effort to provide “observe-orient-decide-act” based decision-making systems in naval and airborne command and control. In this thesis, an infrastructure has been presented using a generic test bed for decision support and the data fusion applications that have been developed for both military and civilian purposes.

Using Multi Agent systems in multi-tier architecture has been used in [7]. It implements an “Evidential Reasoning Network” architecture that is used to integrate various classifiers in a smooth gradation using an “extensible belief algebra framework”. Subjective logic is used when opinions of different agents are fused to create singular opinions which mean resolving dimensionality of opinions from the point of view of the current software agent. The resource [8] which introduces consensus and discount operators has been used as the basis for subjective logic. Further [7] introduces a three layer agent hierarchy that performs data fusion that can be roughly mapped to the JDL Data Fusion Model to be discussed in the next chapter. However, the human sensor angle has not been used which is an important addition to the efforts behind this work.

Another related work is documented in [9]. This paper presents an adaptive human sensor model. A human sensor array within a larger sensor bed of heterogeneous sensors is conceived and the observations of humans are then probabilistically fused with the information extracted from robotic platforms. Their model maps Human Observation Reports into a probability map using Bayesian Network Techniques. Some of their inferences are of direct importance to this work since certain facts are confirmed; like the fact that humans present an adaptive system when
it comes to sensing an event which indicated variable parameters in the human probability model. Bayesian Networks are relied on when general conditions of dependence and independence between events exist that need to be mapped in order to predict the most probable causes and subsequently future possibilities. Finally, [9] mentions the use of the JDL Data fusion model in predicting certain events; generally most architectures can be mapped by function to the JDL model. This is due to the fact that the JDL model, instead of defining “how to” defines “what to” which gives implementation flexibility. Studying the JDL model gives the information necessary to develop one’s own fusion system using different techniques.

Finally, in [10], a framework for multi-source data fusion is presented. A mathematical proof is provided for extracting information from sources with different probability models.
Chapter 3

Background Study

The Joint Directors of Lab (JDL) Data Fusion Model

Historically, the motivation for developing data fusion systems has come mainly from military applications funded by the government. Automatic target recognition (ATR), Identification: Friend / Foe or Neutral (IFFN), threat assessment and situational awareness systems were the first systems that were conceived and developed to assist complex enemy threat assessment procedures.

The Department of Defense applications led to the development of what was called the Joint Director of Laboratories (JDL) Data Fusion model [11]. The JDL model, since its inception in 1991 and major revision in 1999 [12], has become one of the most important frameworks on which many of the contemporary data fusion systems are based. The model describes six levels of processing that are summarized in Figure 3.1. It needs to be stated here that this model suggests methods that need to be used to characterize the phenomena or entities being sensed. However, it is not necessary that these methods or processes follow a fixed order of execution. In fact, Level 4 fusion, as described ahead is a meta-process that needs to run simultaneously to try and gauge the performance of the fusion system.
Level-0 Fusion

This is the data or source information preprocessing stage. Modern sensors have the capability to observe complex processes which can lead to one or both of two things; either the input has a lot more information than needed for a particular implementation and/or that there is a considerable amount of noise or disturbance in the system. These scenarios need to be dealt with before any real fusion starts. Level 0 fusion involves processing raw input data from sensors to prepare the data for subsequent information extraction. Some examples could be image pre-processing to remove noise or irrelevant information, signal pre-processing like filtering to eliminate noise or scale transformations to homogenize the nature of data being collected from different parallel sensor arrays. This is a basic requirement so that the system becomes robust towards random unexpected data.

Figure 3.1: The JDL Data Fusion Model (Hall and McMullen [5])
Level-1 Fusion

This level now seeks to refine object characteristics. We can think of an object as a physical object like a car or a condition or circumstance that we are interested in. Once an array of multiple sensors or sources is deployed in the field, and the data pre-processed, it is obvious that the next stage would be to obtain the most reliable estimate of the object’s characteristics that would most uniquely identify it within proper context. This stage then, creates understanding about entities being analyzed and the geographical conditions they dwell in if physical in nature.

Level-2 Fusion

This stage is also termed as situation refinement and this level uses the results of level 1 processing to identify the relationships of the above mentioned objects with their surroundings as shown in Figure 3.2. This can be thought of as mapping objects to their environment to isolate abnormal behavior. The interaction between objects within the context of their inter-relationship is critical to understand the situation as it unfolds in time. For example, if an animal is the object being observed then depending on how dangerous it is to humans and other animals in its vicinity, various actions might be taken in case it is found in geographical locations of concern.

This level uses techniques like pattern recognition, artificial neural networks and advanced contextual image processing.
Here, information fusion actually occurs at the pattern recognition and classification stage. Pattern Recognition can be done using Neural Networks also. The input feature vector after dimensionality reduction can be changed using the neural net into a classification or identification. In this case the knowledge about classification is "learned" from training data. This process can also be augmented by something known as "Explicit knowledge bases". When such information is used, the scenario is called Context-Based Reasoning. There are many limitations with contextual analysis, especially for new situations and situations involving multiple entities and the way they interact with their immediate environment; especially considering the fact that such relationships change nearly randomly over time. So methods that allow incorporation of explicit knowledge is used along with implicit reasoning. The most commonly used methods to do contextual reasoning [13] are:

1. Logical Templates: Logical templates represent explicit knowledge by describing the conditions under which any object, its condition, activities, related events or general activity can be declared. This representation is characterized as "templates" which must be satisfied. A logical template then specifies logical
conditions that would be necessary to describe or characterize general entities and their behavior in a particular context. Consequently, all other behavior would be considered erratic.

2. Rule-Based Systems: Rule-based systems access what is termed in [13] as a "knowledge-base" of pre-established rules of the form ("If (condition: evidence for M), then Do N"). The rule-base is created by a knowledge engineering process by domain experts. The rules are encoded in a specified format which can allow expressions of probability/uncertainty as well as uncertainty in the resulting consequence. The knowledge base of rules are processed using a search engine that processes incoming hard sensor or human (soft sensor) input data to determine which of the rules in the knowledge base are satisfied and then execute the rules, which changes the dynamic data and required re-search of the rules. This process continues until either a conclusion is reached or there are no more rules available.

3. Bayesian Belief Nets (BBN): A BBN is a graphical representation of uncertain and causal relations in the problem set [14]. It is based on the Bayes theorem for conditional probability. A BBN is composed of nodes and arcs. Nodes represent uncertain variables and arcs represent direct probabilistic influences among the nodes or events as they can be described in suitable context. This means that the arcs indicate the probability of one event following the other. The absence of an arc between two nodes means that these nodes do not influence each other directly and hence are conditionally independent. The following Figure 3.3 shows
a graph structure or topology of a BBN where the ovals represent nodes and the arrows represent arcs.

Mathematically, Bayesian Principles can be defined as follows: A Bayesian network \( N \) is an annotated acyclic graph that represents a Joint Probability Distribution over a set of random variables \( V \), [15]. For this application, any observation report from a Human Observer that provides context to objects can be considered a random variable. So a Bayesian Network is defined by

\[
N = [G, \Theta]
\]

Where \( G \) represents the directed acyclic graph whose nodes are the random variables and edges in the network represent relationships between these variables henceforth providing context. This network then is a representation of independence / dependence assumptions between these random variables and their non-descendents given that \( G \) contains the variable’s parent features.
$\Theta$ represents the values for,

$$\theta_{x_i/\pi_i} = P_B\left(\frac{x_i}{\pi_i}\right),$$

For each realization $x_i$ of $X_i$ conditioned on $\pi_i$, where $\pi_i$ are the parents of $X_i$ in $G$.

$B$ defines the following Joint Probability Distribution over the random variable space $V$:

$$P_B(X_1, X_2, ..., X_{n-1}, X_n) = \prod_{i=1}^{n} P_B(X_i/\pi_i) = \prod_{i=1}^{n} \theta_{x_i/\pi_i}$$

This means that if the current state or condition represented by the variable $X_i$ has no parents, it means that it does not follow any other situation or state in time. In other words, it is unconditional. If it does have parents, it follows the parent conditions and is termed as conditional. Also, the terms latent and observed are used to describe whether the variable represented by any node has been perceived or not.

Level 2 Fusion outputs are the inputs to Level-3 Fusion Sub-Systems which will be elaborated next.

**Level-3 Fusion**

Level 3 refers to threat refinement (if any) and impact assessment (as perceived by level-2) – This can be explained with a concrete example. Any major event is usually preceded by various smaller events that lead up to it.
For example if a car breaks down, there are actually various smaller events occurring simultaneously or sequentially that ultimately lead to the breakdown. If a sensor array in the field perceives these smaller events, then there needs to be a stage in our data fusion model that can predict with a certain associated probability level as to what the future of these smaller events would be. The question that is asked at this level of fusion is: Could the event being perceived lead to a potentially hazardous situation for human beings or any of their systems? This usually means deriving relationships between events and understanding how one event can affect the occurrence of the other. Figure 3.4 explains the idea:

![Figure 3.4: Level 3 Data Fusion Schematic](image)

This prediction binds currently occurring phenomena via a contextual and perception based inference to the model. Inferences about current events and prediction of their future are the key to identifying preemptive measures that might potentially prevent disasters from happening. This stage, thus, requires a lot of simulation, prediction and modeling apart from the machine intelligence fused into it.
**Level-4 fusion**

Level-4 fusion is the process refinement/resource management area which is an important aspect of any system. Conceptually, if the performance of such a fusion system needs to be improved, a good way of doing this would be to give it some sort of adaptive capabilities based on feedback from the existing system. Hence, Level four defines a meta-process that monitors the on-going data fusion process at the other levels and seeks to make the fusion process better by proper resource allocation and sub-system utilization. With numerous possible ways of solving the same inference problem, a constant monitor and upgrade process needs to be in place to constantly improve the system. Optimization in the form of process refinement and control forms a crucial step that ensures optimal resource utilization and maximum throughput for the entire system.

**Level-5 Processing**

Level-5 was one of the latest additions to the JDL model and it introduced the concept of human computer interaction and cognitive refinement. Humans can not only add important information acting as sensors but can also monitor and perceive the system via advanced visualization techniques. This process optimizes system-human interaction. For example consider a closed circuit security camera system with a hundred cameras and 5 displays. The lesser number of displays coupled with the limitations of a short human attention span, makes it necessary that the fusion system needs display only the most important video feeds; importance measured in terms of context. This would involve the system making recommendations and so warrants usage of tools such as
cognitive aids, search engines and human computer interaction interfaces. Sonification (depicting information as sound) of data, 2 and 3 Dimensional Geographical displays and use of Haptic (touch) interfaces form the cutting edge of human interaction with the system.

**Participatory Sensing**

The evolution of distributed intelligent sensor networks has spawned another new research area broadly referred to as Participatory Sensing [16]. The premise this field relies on is the fact that handheld devices used by the general public have gradually become powerful computing machines with various sensors tethered that form a key interface to the immediate surroundings humans are in.

This really furthers the idea that a mobile, ever changing, sensor network can be designed that directly maps the immediate occurrences as perceived by humans themselves. This is quite relevant in the sense that it aims to directly map and study environmental impact on humans. Thus, in this form of sensing, users make up the sensor nodes in that each user controls a mobile sensing device and gathers data which is shared among the general Diaspora.

Generally speaking then, participatory sensing allows people carrying everyday mobile devices to act as sensor nodes and form a sensor network with other such devices or by using existing infrastructure like the internet. A large number of latest generation hand held devices like mobile phones, Personal Desktop Assistants (PDA), laptops and even automobiles are equipped with sensors of the likes of accelerometers, pedometers,
GPS receivers, cameras and audio recorders that form a good combination of sensory devices for participatory sensor nodes. The schematic in Figure 3.5 demonstrates the basic idea behind Participatory Sensing.

![Participatory Sensing Schematic](image)

Figure 3.5: Participatory Sensing Schematic

Mobility of the nodes within the immediate areas of concern is a huge advantage of this theory. Another advantage is that given an optimal power management solution, the excessively strict and careful consideration of power consumption is eliminated. Such a widely distributed sensor network would really help in analyzing wide scale phenomena that are difficult to gauge at a lower geographical scale. Parallels can be drawn between participatory sensing and using higher dimensions to understand lower dimension problems as routinely used in physics and mathematics. Participatory sensing helps connecting seemingly disparate situations and helps discern wide impact phenomena.

One question regarding the motivation for people to participate definitely arises out of this discussion. This motivation will come from the potential of such a sensor network to answer peoples’ daily queries in return for information. User requests for
specific data or information from specific clusters or user groups that might be location
based, at specified dates and times can be answered by participants or data sets that meet
those requirements. These types of requests do not have to be limited to human-mobile
devices but can also include a large set of already deployed and managed static sensor
networks which allow users to freely query them. So people can form their own
participatory sensor networks or be part of a hybrid sensor network which is made up of
human-mobile sensor nodes and static sensor nodes which allows one to gather a richer
set of results. An optimistic vision is to have so many sensor nodes that the sensor
network would become a real-time always-on network which would represent the
physical world very closely. This network could then be queried for various phenomena
in one's locality and return rich sets of results with minimum delay.

Various aspects of participation in such networks are yet to be explored
particularly in the area of privacy concerns. E.g. if a mobile device can be associated with
a person, it can be queried for sensing particular physical phenomena and assuming the
mobile device provides geo-coded data a person can easily be tracked and monitored.

Human sensors have certain characteristics that make them a very non-optimal
sensor in certain scenarios yet quite useful in certain others. There can be quite a few
problems in the full utilization of the optimal use of soft sensors when the devices used to
communicate are cell-phones or hand held personal assistants. Firstly, as already
mentioned, any user would first weigh the benefit of reporting events against cost. Cost
here would be a function of wear and tear, bandwidth utilization fee, if any, and battery
power. Secondly, with the physical limitations of a human, monitoring the status of any
event on a long time basis is impractical. Further, events that are hazardous to the human body would be practically out of reach in case of unassisted and untrained people. Also, there is the geographical problem of coverage with cell phone signals. These issues, other than the emotional, psychological and analytical transients across the H-plane, make humans a difficult sensor to model.

At the same time, soft sensors (some using traditional sensing to report their data) present some unique capabilities. The first one is redundancy. With billions of cell phone users worldwide, the sensor array is so vast that there is a lot of redundancy in the system that alleviates some of the problems of reliability and accuracy associated with soft sensors. Consider a scenario where a live extreme event is being observed by an ad-hoc community of human observers equipped with the latest generation communication devices and portals. With a single human borne sensor, the probability of correct information is far less than when a group senses it. In traditional sensors, the study of sensor placement and actual deployment takes a lot of project effort. But in this case that is reduced to simply extracting and estimating the best estimate of the situation from the various input streams.

There are several characteristics that make humans good for sensing. The first one would be their ability to adapt to a situation. As situations pan out, humans have the capability to change their sensing strategy so that the data sensing operation is not obstructed. Changing stance or choosing evasive action are just two examples of how humans can control hard sensors in a much efficient fashion to maximize the amount of information within the sensed data stream. Secondly, humans are good at gauging human contextual data. Small digressions from the normal human behavior are easily perceived
and understood by humans. Doing this within context of the event is a monstrous problem with machine learning and traditional hard sensing. Furthermore, humans can think about the situation and ask questions based on the reasoning that has already developed from observing the world.

Once a human equipped with a sensor has made an observation, he or she has many options through which to report it. The development of social networking and micro-blogging portals have added to the traditional way in which humans usually report. Another aspect which has not been fully explored is how easily participatory sensing can be implemented without disturbing the flow of everyday life for the participants using their current mobile devices. For example if participants' mobile phone would send data on regular basis through its Wi-Fi interface, and as a result draining the battery quickly, users would find it very inconvenient and a disruption to their life. Another example would be where the users were charged for data transmission through their mobile phone handsets. This is only an example attempting to highlight the point that users' priorities must be respected. So a solution to this arbitrary problem would be to use the mobile phone handset's wireless connectivity facility to send data only upon acquiring a free internet connection so no extra charge would incur and also only attempt this when the connection is obtained through the Bluetooth interface of the device, in which case the battery consumption would be much lower compared with a Wi-Fi connection. Additional layers of security would be absolutely essential to maintain privacy in such cases.

The vision for participatory sensing applications is to allow users to go on about their everyday life while performing data collection tasks without any extra effort. Also
with necessary means available to users, they should be able to deploy their own participatory sensing applications and encourage other people to join their network and create a community based sensor network. These networks could then be queried for data, perhaps through internet.

**Multi Agent Systems (MAS)**

A multi-agent system (MAS) is a system composed of multiple interacting, independent and in many cases highly distributed pieces of software entities called ‘agents’. These software agents can be called *intelligent* if they have the capability to allocate resources and act independently on an evolving situation aimed towards meeting the broader goals of the entire system, of which they are an integral part.

Multi-agent systems can be used to break down or divide complex problems into simpler sub-divisions that can be better tackled in a distributed and parallel fashion. Examples of problems which are appropriate to multi-agent systems research include security threat analysis, online trading [17], disaster management and response [18], and modeling social relationships and structures. For a more extensive discussion, see [19].

The intrinsic functionality of an agent can be broken down into sub-functions as shown in Figure 3.6. The Percepts are the perceived form of the quantity being sensed by the sensor and can be expressed as the presence or absence of the quantity itself. The agent analyzes these inputs and decides on the action to be taken based on the initial parameters in the learning module. The Learning module itself has a handle to the intrinsic performance parameters of the Agent.
A time dependent learning phase controls the output of the Agent. Based on the learning phase, the Performance Monitor can be adapted to the changes perceived to better meet the desired goals of perception initially decided by the designer of the Agent. Hence, any recommend the agent makes is a function of the Perceived inputs as well as the Knowledge of the Process gleaned via statistical analysis of the data. The recommendation can be used to control another mechanism or be the input to another Agent that uses these recommendations as its Percepts. This flexibility in design that can
also utilize a hierarchical structure is a key benefit of using the Multiple Agent Type Approach to a particular problem.

For this particular application, the role of each agent is to perceive the environment via soft and/or hard sensors and then create knowledge about that same environment. The sensor output can be encapsulated from the agents also in the sense that it is not necessary to implement the Agent approach to extracting information. The design should be such that the MAS layer should be entirely swappable by a different implementation. However, for various other reasons make a MAS based approach ideal for the purposes of this work. These are the main advantages:

1. Parallel and Distributed Functionality: For example, consider a scenario in which there are multiple processes that occur simultaneously. The sensors that measure these events might be of different types and capacities and might border new types of sensor concepts such as the World Wide Web as well as independently reporting human beings. To evaluate such a scenario, various input data streams have to be made sense of. Further, the system might take the help of various resources other than the sensor feeds. Thus, a highly parallel and distributed system is needed that can, with a good degree of robustness solve the problem.

2. Processing Speed: For a system that is being designed to predict extreme events using advanced data fusion techniques, it is imperative that the correct results be achieved in real time. As a direct response to this requirement, one of the other major advantages of using a multiple agent based system is increased speed of operation which directly stems from the fact that the system is distributed and
highly parallel. Furthermore, the parallel nature of MAS can help alleviate the shortcomings imposed by time-bounded reasoning requirements as in the case of real time data fusion and event prediction.

3. Scalability and Modularity: Like mentioned previously, another benefit of Multiagent systems is their scalability. The system can be designed as being modular with each module giving a particular ability to an agent; this should make it easier to add new agents to a Multiagent system. This would save design effort as well as improve reusability of code and logic. This also means that any System that evolves over time and needs updating can be updated with relative ease.

4. Robustness due to Redundancy: robustness is a benefit of Multiagent systems designed to have redundant agents. If control and responsibilities are optimally shared between redundant agents tasked to carry out the same function, the system can tolerate more failures as well as incorporate for randomness in the entity being analyzed.

5. Easier Maintenance and Monitoring: Besides all the advantages outlined already, it is easy to see how such systems are easier to maintain and monitor. Further, modifications to the system are also easier to incorporate.

An example of multi agent architecture is the DARPA Cougaar project [20]. This platform can be used to create a robust society of agents that can solve any contextual problem like for example developing a Human (Soft) Sensor array over the I-Space. A Discussion on the attributes of I-Space follows.
I-Space Introduction: Internet as a Data Source

The internet has evolved into a huge repository of information that is updated in real time by millions of users worldwide. It defies geographical and political borders and has become an important tool for people to interface with each other. With the advent of the blogging and micro-blogging culture, people around the world exchange millions of gigabytes of information every day. However, not all of it is useful from a serious analysis standpoint. This raises the first major question. How should the useful bits be separated from the useless? This question can only be answered from a contextual viewpoint. In other words, depending on the context certain information can be relevant or irrelevant. It is safe to assume that those are not the only two classifications of information in the case of multisensor data fusion. This is because different information pieces might have indirect relationships that might be difficult to perceive at the superficial level. So, essentially the web is like an interface between a huge soft sensor pool and the data fusion system.

The other dimension to I-space is the presence of S-Space data in the form of database repositories or any other data storage or transmission system. Any public domain information supplied by governmental sources might be characterized on the basis of the sensor it originates from. Thus, the I-Space cannot be simply defined as being a particular type of sensor system. Hence, it needs to be understood as a universal interface with different sources both hard and soft in nature.

It is easy to see what the problem is in this case, sensor characterization is difficult where humans are the primary data origin elements, and if the web does in fact
act as a source of hard sensor data, authentication and trust becomes a challenge. Some of these questions will be answered further in this report.

So, since we have already established the fact that the internet can be used as a potential source of data both from hard as well as soft sensors, here is an example showing how this can be done. The Figure 3.7 shows a conceptual framework that can be used to map a certain domain of the internet called Microblogs.

Microblogs really became popular and mainstream with the advent of the web service twitter [47]. The whole idea of a limited amount of information being transmitted by each user in each interaction with the blog opened a whole new world of real time information. This information is essentially a pool of reports from users or un-mapped soft sensors or other automated hard sensor sources. They have been termed un-mapped in the sense that any notion of a basic trust level or reliability of these sensors is absent at the outset.

A source of user generated hard sensor information is images and videos. In this case, the environment is sensed with traditional S-Space sensors like video cameras and still cameras by human beings. Flikr is an online image content sharing service that has a programming interface for external applications to interface with public domain data.

The figure thus represents a basic web model that uses Twitter and Flikr as the human web based data sources and combined with any other specialized information, can be used to add meaningful contextual information to the fusion model.
The central block shows a Multiple Intelligent Agent Society that can be developed to make sense of all the information being extracted from, for example, Twitter over the internet with its REST API (Representational State Transfer Application Programming Interface). Here is a short description of the Data Sources from figure 3.7:

1. Microblogs: Microblogs like Twitter can be used as a platform to create sensor arrays comprising of humans with particular attributes. These attributes can be chose depending on the situation being analyzed or information being sought. The user messages can be viewed as live sensor feeds. The motivation for using Twitter will be discussed ahead, however a primary real life indicator of the
technology’s use was observed during the Mumbai Terror Attacks in 2008. According to the Telegraph (UK) “Messages, known as “tweets”, were being posted to the site at a rate of around 70 tweets every five seconds when the news of the tragedy first broke, according to some estimates,” [21]. With a large distributed network of human beings that understand their environment, such a tool can be envisaged as a real time indicator of the happenings around the globe.

2. Semantic Web: Services such as Friend of a Friend (FOAF) [22] that intend to help create a semantic map of human related information can then be utilized to extract more information about the soft sensors of interest. This step is geared towards creating a basic trust map that can somewhat predict the behavior of human sensors.

3. Specialized local Web Information: For any organization that aims to create this level of cognition within an in-house system, there can be other web connected resources that can be utilized to get more information about soft sensors. Examples include any local directory listings, wiki entries, project databases or any other project or application that can help augment the nature of sensor feeds to the system. A heterogeneous array can be developed aimed towards creating contextually aware fusion systems. Statistical analysis can help map soft sensor reliability as well. The system then can make decisions related to context and can feed that knowledge to the traditional data fusion system that can then use it to come to a quick and more accurate decision.
Chapter 4

Architectural Requirements for enabling hard and soft Data Fusion

This chapter describes the basic cyber infrastructure that is aimed towards fusing hard and soft sensor data. Considering the concepts already discussed in this thesis, there is a need for developing a model that can effectively help fuse data from different sensor arrays effectively. When we mention a Heterogeneous Sensor Array, we mean that the Array might contain one or more types of S-Space, H-Space and I-Space Sensors. Based on the characterization of these sensors, each sensor type contributes a certain weight to the overall decision calculation. Based on the decision threshold the weight results in a recommendation. Besides mathematical trust values for H-Space and I-Space Human Reporters, there needs to be another mechanism to convey characterization values since these sensors have subjective characterization tags, hence the need of Complex Human Computer Interaction Layers above the actual logic layer.

Functional Architecture

For this thesis, three sensor types are considered to construct this Architecture. Namely, Networked Sensors that include for example Closed Circuit Video Surveillance Systems, Ad-Hoc Sensor Networks and any other Fixed Topological Sensor Networks like Pressure Sensors or Fixed Position Microphones; Human H-Space Arrays of Observers and I-Space Human Reporters that use channels like Microblogs to publish public domain information online. The JDL model forms the basis of this architecture. Pre Processing and Entity Recognition are the obvious first steps. Metadata development
or annotating is another step that helps categorizing the information in the data at a very basic level. Subsequent analysis includes situation awareness using pattern recognition and then prediction. Finally Human computer interaction is used for proper transfer of results.

Figure 4.1 shows an Architectural Framework for data fusion from I-Space, S-Space and H-Space sensor arrays. This architecture is the result of the research of Hall et al [23], [24]. This is the primary architecture that will be developed upon and added details to in this report. As mentioned, this architecture is based on the JDL Data fusion model.

Data from the world is fed to the pre-processing module via sensors. Pre-processing removes much of the noise from the data at this stage. Specialized tools can be used at this stage to demonstrate the implementation of these pre-processing tools. For example, for image and video processing, MATLAB is a good research tool. Intelligent Agents are crucial for I-Space Sensor Array formation as well as primary data extraction.

After primary synthesis of a clean dataset, metadata extraction and semantic labeling is performed to link the subjective data via specific contextual relationships. This forms the input to the higher layer JDL Data fusion layers. The Higher level fusion layers are dynamically tasked. However, the sensor array formation and preprocessing stages can also be dynamically tasked as will be presented in this work. The word dynamic in that sense would mean creation of sensor arrays on the fly on the basis of the panning situation and need.
Dynamic resource allocation is a constant feedback process that is used to monitor proper functionality of the system. Modeling and simulation forms a crucial monitoring and optimization tool at each step. Live simulation examples of I-Space Twitter based Human sensor array formation has been demonstrated in this thesis. Detailed mathematical modeling for the system can be found in [25]. [26] Presents a Semantics-Sensitive Integrated Matching Algorithm for Picture Libraries that is relevant to Metadata Generation for image data and [27] has an in-depth analysis to understand how to improve the Data Fusion Process using Semantic Level Whole Brain Analysis. These examples are helpful in understanding Semantic tagging of subjective information and the subsequent metadata generation.
After analyzing the above architecture, there was a need to develop the details of the system in areas such as I-Space human sensor array formation using multiple agent systems, system communications interface architecture to link more than one unique systems, as well as development of S-Space image and video processing capabilities that would prove the feasibility and efficacy of the above design. Thus, in the next chapter new architectural details have been developed and demonstrated using different implementation algorithms and across various platforms.

**Soft Sensor Characterization**

Human sensors have certain characteristics that make them very non-optimal as sensors in certain scenarios yet quite useful in certain others. There can be quite a few problems in the full utilization of the optimal use of soft sensors. The first problem is the creation of Sensor Arrays specific to sensing a specific part of the human landscape. Each sensor node in a traditional hard sensor array has the same characterization which produces homogeneous arrays in terms of behavior. However, in case of human sensor arrays, each human will have different characterization due to individual qualities and capabilities. The key then is to identify a common trait or characteristic that would provide the common ground to have them all as part of the same array. In other words, a relationship map needs to be developed. This is discussed in this and subsequent chapters.

Secondly, as already mentioned, in case of participatory sensing type scenarios that are expected to create human observer arrays, any observer would first weigh the
benefit of reporting events against cost. Cost here is a function of wear and tear of equipment, bandwidth utilization fee for reporting as well as physical resources of the equipment like battery power that will now have to be shared between public and private use.

Thirdly, with the physical limitations of a human, monitoring the status of any event for a controlled time basis is impractical. The word “controlled” is critical since monitoring the tasking of human observers whether over I-Space or H-Space has limited practicality in the civilian domain. Defense force personnel can be controlled more in this case. Further, events that are hazardous to the human body would be practically out of reach in case of unassisted and untrained people. Also, there is the geographical problem of coverage with cell phone signals for example. These issues, other than the emotional, psychological and analytical transients across the H-plane or humans in the I-Space, make them a difficult sensor to model.

At the same time, however, soft sensors (some implementing traditional sensing to report their data) present some unique capabilities. The first one is redundancy. With millions of cell phone and internet users worldwide, the raw sensor array is so vast that there is a lot of redundancy in the system that alleviates some of the problems of reliability and accuracy associated with soft sensors. Dynamic arrays can be created with enough redundancy to help collect statistically relevant data for decision making. Consider a scenario where a live event is being observed by an ad-hoc community of human observers equipped with the latest generation communication devices and portals. With a single human borne sensor, the probability of correct information is much smaller than when a group senses it. In traditional sensors, the study of sensor placement and
actual deployment takes a lot of project effort. But in this case that is reduced to simply extracting and estimating the best estimate of the situation from the various input streams.

Apart from this, there are several characteristics that make humans good for sensing, one such characteristic being the capability to adapt on the basis of the changing circumstances. As situations pan out, humans have the capability to change their sensing strategy so that the data sensing operation is not obstructed. Changing stance or choosing evasive action are just two examples of how humans can control hard sensors in a much efficient fashion to maximize the amount of information within the sensed data stream.

Further, humans are good at gauging human contextual data. Small digressions from the normal human behavior are easily perceived and understood by humans. Doing this within context of the event is a very difficult problem with machine learning and traditional hard sensing. Furthermore, humans can think about the situation and ask questions based on the reasoning that has already developed from observing the world.

Once humans equipped with a sensor have made their observations, they have many options through which to report them effectively and in a timely fashion. The development of social networking and micro-blogging portals have added to the traditional way in which humans usually report.

**User Generated Content: I-Space**

The research team at the Center for Network Centric Cognition and Information Fusion at the College of Information Sciences and Technology at the Pennsylvania State University uses the following definition of ‘User-generated’ content: “a blanket term that
describes such tools as blogs, wikis, tagging, and uploaded photos and videos at sites such as Facebook, Flickr, and YouTube.” [28]. This implies that users are free to upload content within the boundaries of public decency. This data might be targeted towards a select group of people usually within the sphere of influence of the person in question. But, at the same time, it might be for a more general audience like that demonstrated in the micro-blogging portal twitter. The development of Twitter has lead to an explosion in reports that convey the state of circumstances in the person’s immediate vicinity. These independent reports can provide a lot of information about globally occurring events that might not be perceivable at a smaller geographical granularity.

However, the biggest issue with social media is that since it is not produced with any fixed or strict standard, a) it can be in any format, b) it can vary from being entirely accurate to complete fabrication and c) the reliability of data cannot be conveniently modeled because of the absence of any fixed pattern in human behavior where human is assumed to be an untrained being.

However still, there are certain aspects of it that do make it a topic worth researching:

1. Quantitative Scale: The amount of data being uploaded is phenomenal. This means that given the right tools that can sift through and look for that critical bit of information; this can still become a crucial value addition to our overall understanding of any situation.

2. Geographical Extent: Many of the web powered repositories of user fed information like Wikipedia are available in languages that span the entire globe.
With the free availability of the internet in many parts of the world, it has become easier for people around the world to share information.

3. Impact Assessment: One of the best features that is a function of both the above aspects is the fact that given the wide scope and data availability, wide spread patterns of phenomena can be statistically derived by implementing pattern recognition systems. This means that combined with the available hard sensor data, this information can help identify the highly distributed effects of the same event. This can help gauge the impact of these events at the global scale.

4. Aid to Machine Learning: The Web is full of instances where humans decide on the basis of their knowledge of human related idiosyncrasies that is very tough to infer via machine learning. Examples include decisions about sellers on e-commerce websites like Ebay and Amazon and websites like Yelp for example [29]. But since there is a lot of tagged information on the web and this feature coupled with the introduction of the Semantic Web, will make it a lot easier for us to create richer data sets that machines can be trained on.

Traditional data fusion can improve an extended sensor system’s performance in four ways:

1. Representation: Parts can convey a sense of the whole. This means that seemingly unrelated events can actually be part of a pattern and can lead to the detection of future events of concern.
2. Certainty: Good fusion practices increase the probability of a gathered piece of data

3. Accuracy: Good fusion practices reduce noise and errors

4. Completeness: Adding new knowledge to the current environment can give more information about the event within proper context.

**Human Observation Reports**

The design of human observation reports is a research area in itself. Depending on the system that is involved in extracting information from these reports, it is crucial to design a format that helps speed the process. System complexity will increase if no report format is chosen. Here lies a paradox. The number of human observers willing to comply with the requirements of the report format will decrease with an increase in the requirements themselves. So by design, there needs to be the built in capability within the system to parse and understand unformatted and formatted observation reports.

Broadly classified, human observer reports can either consist of hard sensor data from traditional hard sensing resources or textual and voice records. Reports of the former type contain information embedded in a format better suited to machine analysis than the latter. As an example, a user reporting an event, using a video camera to record or stream a video feed of the event along with voice annotations sends both types of reports. The Video feed in that case would be a hard sensor report and the voice annotations are a Report generated by the person analyzing the situation on scene.
However with minimum calibration of these sensors within actual user control, pre-processing would be a big step to eliminate noise from hard sensor feeds from human observers.

Battle Management Language (BML) [46] is “an unambiguous language used to command and control forces and equipment conducting military operations and to provide for situational awareness and a shared, common operational picture.” BML was developed by work funded by the US Army’s Simulation-to-C4I Interoperability Overarching Integrated Product Team. While its implementation was geared towards developing a language best suited for machine analysis (using software), it was also geared towards army usage. However, it is based on and an extension of the 5 Ws, the Who, What, Where, When and Why. The type 2 Chomsky Hierarchy grammar [30] also otherwise called the context-free grammars was chosen for BML since it is constrained enough to be supported by automated processing as well. Using these basic rules from an already well developed Language, a civilian centered simpler Language structure can be developed that can be used by Human Observers to report event descriptions via text. This is most helpful for Human Observers using the I-Space to transmit information specifically to the data fusion system. This would make the most important Reports, the easiest to analyze. The rest of the Public Space Reports in the I-Space will most definitely have more of the Chomsky Hierarchy type 1 or Context-Sensitive Grammar which will require complex language processing to analyze.
Chapter 5

A Prototype Cyber Infrastructure Environment: I-Space

Design and Implementation

The blogging revolution has brought about a huge change in the quantity of information created by users perceiving their surroundings. Users can be objectively classified by identifying a *training* level which ultimately contributes to the level of trust. The level of training varies in the H-space from none to very high with respect to a particular situation and depending on context. For example, a common user might be not trained enough to identify the causes of a fire but a trained firefighter is. So from the point of view of an H-Space sensor reporting a fire, the latter has had more training than the former and so will garner higher initial trust. Similar H-space sensors with proven training records can contribute a lot towards understanding a situation that otherwise might be difficult to perceive from the S-Space standpoint alone.

As previously mentioned, Microblogs like *Twitter*, with their easy interface and public data have created a potential source of such H-space sensors. The question of how to create a sensor array from this raw pool of H-space sensors will be answered by this demonstration. *Twitter* users can be mapped to an I-space human sensor array and subsequently used as sources of human reported data.
Principle

Humans, or simply, users, control input to their own micro blog. They use their sensory organs to report observations. This scenario can be understood within the traditional context of a sensor array with N, I-space human observational sensors and their corresponding N input data streams. So in this instance, each user is a sensor node and the respective blog time line can be considered as a live textual sensor feed. Depending on the nature and current state of the human sensor, the probability of correctness or accuracy of this data greatly varies and is very difficult to predict. This would require long term analysis for validity. The goal of this segment of the experiment is to show how intelligent agent based systems can be used to perform some of the initial tasks of building a sensor array from a vast array of unmapped sensor resources.

General Design Decisions

To address immediate design requirements, a proper cyber framework needs to be chosen that simulates a raw resource pool of human observers. Once that is done, a model can be built that uses this currently existing public domain infrastructures to create the desired dynamic array to facilitate receiving human observation reports through the I-Space.

The following figure; Figure 5.1 shows the architectural schematic of this design. Within I-Space, there exist many web services and applications that already have a huge User base that inputs real time information daily. As already mentioned, Microblogs are one type of such web applications and picture sharing sites like Flickr for another. Given
that a suitable Application Programming Interface exists one can derive textual and Image information from these Applications and use them for processing.

**Figure 5.1: I-Space Applications Interface**

The Microblog Twitter is chosen to represent a raw pool of H-Space sensors. This decision is based on the following reasons:

1. Usage (Quantity of Users): According to Quantcast [31], roughly 4 million users visit the blog everyday within the United States alone. The sheer quantity of H-space sensors, and the fact that statistically a very significant number of twitter users used the blog to transmit information, make it immediately apparent that a lot of information can be gathered in a short while, which is ideal for quick development and testing.
2. Public Data Content: Due to the fact that a sizable percentage of all users make their tweets public, it is easy and above law to reach out for this “public domain” information and use it for research and development of information based knowledge.

3. Programming Interface: Further, the Microblog public data is accessible to external applications via a strong application programming interface (API). The ease with which the Twitter Java API (Twitter4J [32]), can be combined with any agent architecture typically based on Java like for example, the Cougaar Agent Architecture, also makes it possible to develop code faster. So the system is independent of platform or any particular architecture.

An Official PSU Extreme Events Lab Twitter Account is used as the Host User Account. This Host Account is used to develop the sensor array. This will be referred to as PSUEEL in this report. Now the goal is to create a dynamic array of sensors based on initialization parameters. Figure 5.2 is a representation of the general Infrastructure that has been implemented to demonstrate the creation of I-Space dynamic soft sensor arrays. It also depicts how the S-Space video data ties into the architectural framework of this infrastructure. The human in the loop provides initialization parameters to the system. Based on these, the multiple agent system creates a soft sensor array over Twitter. The agent system can also be capable of changing the initialization coefficients to meet sensor array requirements as shown.
A dynamically created soft sensor array can then be used to collect textual human observation reports to create contextual knowledge about an event or target entity. A deeper discussion on the design of the sensor array characterization layer follows.
Sensor Array Characterization Layer Design

1. As discussed, the software agents help in modularizing a bigger, more complex task. In this case, a distributed Architecture has been used to demonstrate how any similar society can be built. The DARPA Cougaar Agent architecture is a good example that is used to demonstrate the point.

2. The primary goal of the society is to create a dynamically regulated test array of I-space soft sensors based on certain prerequisites. These prerequisites provide the basis of the initial conditions used to create the array.

3. Example initialization parameters are controlled by governing factors like human relationships, location, age, gender, work environment etc.

4. Since the target was the Penn State University Park Campus, the initialization used a list of trusted users from the NC2IF lab.

5. The premise used to develop the sensor array is the fact that users share a certain relationship and mapping these relationships give us the links between the nodes of our cyber mesh network of sensors.

6. The Agent society can be highly distributed which is shown by using the distributed configuration supported by Cougaar.

7. Each agent needs at least one JVM (Java Virtual Machine) to execute. Each JVM can house several Agents. A single hardware computer can have single or multiple JVM running on it or each JVM can be on a separate physical machine thereby proving that this is a truly distributed Agent Architecture.

8. Whatever the physical architecture used, each software agent interacting with
Twitter via its Java REST (Representational State Transfer) Application Programming Interface (API), fetches some information and saves it to a central Database.

9. The nature of information that each agent accumulates is of two types:
   a. I-Space Soft Sensor Characterization: In other words this is information about the type or traits of the sensor itself.
   b. Data: This is the data stream fetched from the Sensor Array.

**Cougaar Software Agent Society Structure**

![Cougaar Society Structure](image)

**Figure 5.3: Cougaar Society Structure**

The Agent Society is comprised of 6 nodes. A “Node”, in Cougaar terminology, represents a single Java Virtual Machine (JVM), which, practically exist on separate hardware systems and house Agents and their plugins; Figure 5.3. However, for our case, each node is simulated on the same physical machine. So, one physical machine supports 6 JVM instances in this case. This means that system resources are divided between each
VM. From the practical standpoint, to improve efficiency, these would be deployed on 6 different physical processors.

Each Node has the same internal Agent Structure for simplicity. Two Agents exist in each node. They share a Controller – Worker Relationship. The Controller Agent interacts with the Worker Agent and invokes the functionality implemented therein. The Worker Agent returns a status message indicating success or failure of the implementation within it. The Worker Agent can be made to communicate all of its output to the controller using Cougaar Blackboard Functionality easily. The reason this structure is chosen is to typify a hierarchy of Agents within the Society. So the Controller Agents can be part of another Society that uses the output of the current Society as its input. This hierarchy can then be used to facilitate human sensor classification at different levels. This is termed as a “Layered Soft Sensor Characterization” and is explained later in this chapter. A central database is used as the local data storage repository. Each Worker agent communicates its data to specific tables in the database the schema of which has been discussed in Appendix B. So whenever the Controller Agent sends the Control Signal to the Worker Agent, the Worker Agent interacts with Twitter via the Java REST API and stores the data in a fixed location in the Database. Based on functionality, other Agents use that data as their inputs.

Algorithm Details and Working

To understand the implementation details of this algorithm, it is necessary to be cognizant of Twitter terminology. For a list of these, refer to Appendix A. For a detailed
version of the database schema, view Appendix B. The description below is from the point of view of the worker Agent within each node. The controller, like already mentioned, simply controls when the worker Agent executes. Before the algorithm is discussed, it is necessary to indicate the logic behind this sensor array generation. As already suggested in this report, a relationship map needs to be built between users. The strength of relationships (\( \mathcal{R} \)) between users can be ascertained on the basis of the following characteristic parameters:

1. **Friendship Dimensionality (FR):** If two users are friends, a stronger relationship is indicated than when one simply “follows” the other. At this stage, a positive or negative tag is not associated with any relationships being talked about. The word strong used in the relationship context can either be positive or negative with both cases being important for a fair sensor array.

2. **Replies (Talk Count) \( R \):** If users respond or reply to messages from other users, it indicates a stronger relationship since users directly and publicly choose to respond to messages from other users.

3. **Favorite Count (\( F \)):** Another functionality that Twitter offers is where users can tag messages as their favorite. More the number of favorite messages from a user, more the chances of them having at least a strong one-sided relationship with the other.

The formula to calculate the Relationship Strength \( \mathcal{R} \) between users \( x \) and \( y \) can then be:

\[
\mathcal{R}_{xy} = FR_{xy} + \sum_{n=0}^{N} R_{x\rightarrow y}(n) + F_{x\rightarrow y}(n) + \sum_{n=0}^{N} R_{y\rightarrow x}(n) + F_{y\rightarrow x}(n)
\]

Where both dimensions of a relationship are used (from user \( x \) to \( y \) and vice versa)
Depending on the weight values assigned to the functions FR, R and F; and the value of N, the value of $\mathcal{H}$ can be found and compared to a dynamic threshold.

Hence, a weighted sum of all these parameters can give a value indicating the strength of a relationship and can be compared to a threshold value. Relationships with values above the threshold qualify as “Raw Sensor Nodes” for the Sensor Array. What follow are the detailed steps implemented in the Algorithm.

Basic Design

1. Node 1 Worker Agent: Fetches direct messages to PSUEEL and saves them to our database. Direct messages indicate direct contact between a user and our account. This feature of Twitter can be used to deliver formatted information from users participating voluntarily in reporting events.

2. Node 2 Worker Agent: This agent fetches basic friend information of PSUEEL and saves the friends of PSUEEL (the people followed by PSUEEL) to the database. For the initial run of the code all users of PSUEEL are fetched and saved to the `TwitterFollowedUsers` table in the database. This is so since initially the table is empty and so all friends need to be fetched. Subsequent iterations of the code access a different SQL query that updates user information while adding only new friends. This node hence updates any changes in the Name / User Name fields of any user’s profile which is critical as the system needs to keep track of the user’s changing characteristics.

3. Node 3 Worker Agent: Now that Node 2 has provided a list of friends, the Node 3 worker agent Fetches the friends of the trusted users and populates the `UserTalkRepository` table in the database. It also fetches the timeline and talk
count for each user. The main difference between this Agent and the previous one is that this one not only fetches information about the friends of PSUEEL but also of the friends of friends. This chaining methodology creates the human array that will be modeled as a sensor array.

This agent also starts building relationship maps based on certain pre-requisite conditions. The existence of two dimensional friendships between two users indicates higher relationship strength than single dimensional ones. That is, if both users follow each other, it is a stronger bond than when only one follows the other. This Agent keeps track of this dimensionality of friendship.

Also, it keeps track of each relationship on the basis of talks. A talk would mean a conversation where users reply to each others’ messages. It counts the number of replies to, or references that trusted users make of their friend users that can be added to our sensor array.

4. Node 4 Worker Agent: Worker agent 3 kept track of friendships on the basis of the dimensionality of friendship and talks. Now the Node 4 worker considers Favorites. If users favorite other users’ messages, it indicates a positive liking of the message and in turn a stronger relationship. This agent fetches all favorite messages of trusted users and updates the favorites count for prospective-trusted users in the database. This value contributes to the overall weight calculation. (Along with the two facts fetched by the above agent as mentioned)

5. Node 5 Worker Agent: Now we have three elements that can be used to characterize a relationship between users; namely Friendship Dimensionality, Talk Count and Favorite Count. This worker agent then calculates weights of
relationships between trusted and prospective-trusted users by the Relationship Equation. Each of the three elements has a different weight that adds to a relationship. Thus this agent calculates the total weight as a weighted average of the contributions of the three elements and decides on whether to trust a new user by comparing those values to a preset threshold.

Once users are found to have the relationship weight greater than the threshold and given that they have open profiles, they are added to the raw array of I-Space human sensors and sent friend requests so that they directly become friends of PSUEEL. If profiles are found to be protected, requests are sent users and they are saved to a different table called TwitterWaitList.

6. Node 6 Worker Agent: Based on the decisions by Node 5 worker, if users have protected accounts, this agent monitors whether our account gets authenticated by the new now-to-be-trusted user. In other words, it periodically checks to see if the new prospective friend users approve of the friendship request and it handles accepts and rejects accordingly.

Dynamic Sensor Array Observations

It is observed that subsequent iterations of these independently working agents create a raw sensor array of users with the initial vector of trusted users from within the Extreme Events Lab at Penn State. Now, on the basis of the dynamic input parameters, like location for example, further filters can be applied to distinguish sub-arrays. This is how a dynamic array of I-Space human sensors can be demonstrated to have been
Initialization Location Based Arrays Needed (Classify Sensors on the basis of Geographical Location)

Figure 5.4 shows a distribution of the Sensors as derived from the Observer Metadata from Twitter. The initialization Array of users comprised of Members of NC2IF as mentioned. Clearly, at least 307 Users were identified to be in the USA. This is raw Sensor Characterization data and can be verified by observer feedback. Once observer feedback is received, these values can be periodically updated.

Further classification can be achieved by looking at the USA State Wide Distribution in Figure 5.5 that shows a United States, State wide distribution of Sensors that are a part of the Array Formed. Once again, distribution data is extracted from the Twitter user
data. These raw values represent a general trend in observed location data.

Figure 5.5: Dynamic Array USA State Wide Soft Sensor Distribution

Now in Figure 5.6, city wide observer distribution data within the United States is plotted. These plots are indicative of the fact that on an average, when an array is initialized to achieve location specific human sensor arrays, it is best to initialize the array with a fixed user set present in the same geographical area. It stands to reason that once the raw dynamic observer array of this nature is created, as already mentioned, a participatory sensing system can be used to ask user voluntary involvement within the study and data verification can be achieved. With that, an initial base trust value can be set for each observer that can be updated periodically by testing observer knowledge perceived as responses to specific questions or reactions to different contrived stimuli.
Lastly, the focus can be shifted to time zones. Time zones also can group Human Sensors into a geographically relevant boundary. Figure 5.7 is indicative of the least number of people verified by Twitter metadata. When a system replacing Twitter is developed to finally work as a front end, this should factor into the design of its Information Schema. Location on the basis of latitude and longitude should also be
included to pinpoint the exact location of the sensor.

![Bar chart showing the distribution of sensors across different time zones. The chart is labeled as Figure 5.7: Dynamic Array Time Zone Based Soft Sensor Distribution.]

**Message Exchange Architecture**

Interfacing Agents amongst themselves using the Cougaar Message Passing Features and by other techniques with other external software Entities was critical for proper internal flow of information. A Message Exchange Architecture is developed and implemented. Here follow the details of this implementation:
Agents can exchange information amongst themselves and external applications in number of ways. The following figure shows the major communication links used for the current design.

Figure 5.8: Message Exchange Architecture

1. Cougaar Society Blackboard: The Cougaar Agent Architecture provides direct message passing via a shared blackboard utility [33] which is analogous to the shared memory concept. The obvious merit is quick, efficient and fully controlled message passing between distributed agents.
2. Relational Database Management System (RDBMS): In case where long term data storage and analysis is needed, one agent society can communicate with the other using the central or satellite databases. Relational Databases like MySQL are a very powerful tool for data management. Such a Database can be linked to Cougaar Java programs via the Java Database Connectivity (JDBC) Driver.

3. Although Cougaar uses the distributed approach to message passing, additional interfaces need to be implemented in the system by developing an API. This API not only creates a bridge between the database and external applications but can service local applications as well over the IP Network. Combined with the Java Servlet Technology, this can be a powerful data exchange mechanism.

Servlet Details: To demonstrate how the Agent Society Information can be distributed among external applications without sharing database implementation details, a Two-Servlet Architecture is developed. Here follows the description and functionality of the Interface:

1. “GetContactList”: This Servlet is used to return a list of the dynamically created I-Space Soft Sensor Array Nodes by their most recently updated Name on Twitter. This code snippet shows the basic XML Message Structure returned as response. XML is ideal for message passing between different applications and its structure makes it easy to extract information from. The following code snippet in figure 5.9 shows the XML Structure of the response received when the contact list is requested from it.
Figure 5.9: XML Based Servlet Response Format for Contact Lists

Figure 5.10 shows a response example to a generic request to this Servlet.

```java
response.setContentType("text/xml; charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<?xml version="1.0"?>");
out.println("<ContactList>");
for (i = 0; i < index; i++) {
    out.println("<Contact>");
    out.println("<Name>");
    out.println("<value">+identity[i]+"</value>");
    out.println("</Name>");
    out.println("</Contact>");
}
out.println("</ContactList>");
out.close();
```

Figure 5.10: Servlet Response for Contact List Request

```xml
<?xml version="1.0" ?>
<contactList>
    <Contact>
        <Name>
            <value>zorp75ck</value>
        </Name>
    </Contact>
    <Contact>
        <Name>
            <value>Anna Tarkov</value>
        </Name>
    </Contact>
    <Contact>
        <Name>
            <value>Arnold Aranez</value>
        </Name>
    </Contact>
    <Contact>
        <Name>
            <value>t Kilbride</value>
        </Name>
    </Contact>
    <Contact>
        <Name>
            <value>Glen Turpin</value>
        </Name>
    </Contact>
</ContactList>
```
The Servlet returns a list of I-Space human observers as networked by the Agent Society on Twitter. Detailed public information about any of these Observers can be extracted from the Database by issuing a different request to another Servlet which is detailed next.

2. “CommunicationServlet”: This Servlet returns detailed information about each human observer if the request contains the parameter ‘Contact Name’ chosen from any of the observers returned by the GetContactList Servlet for example. This Contact Name parameter does not obviously need to originate from information extracted by GetContactList. The source of that information can be the Twitter API as well so any external application can fetch user information locally without requesting information via the API. Figure 5.11 represents the code snippet showing XML Structure of this response:

```java
out.println("<Contact>");
out.println("<Name>");
out.println("+finalInformation[0]+"/>
out.println("</Name>");
out.println("<TwitterScreenName>");
out.println("<value">+finalInformation[1]+"</value>");
out.println("<Description>");
out.println("<value">+finalInformation[2]+"</value>");
out.println("<Location>");
out.println("<value">+finalInformation[3]+"</value>");
out.println("<Location>");
out.println("<TimeZone>");
out.println("<value">+finalInformation[5]+"</value>");
out.println("<TimeZone>");
out.println("<TwitterID>");
out.println("<value">+finalInformation[4]+"</value>");
out.println("<TwitterID>");
out.println("</Contact>");
```

Figure 5.11: XML Based Servlet Response Format for Contact Information
An Observation example is shown in Figure 5.12 that represents the response to this request:

```
- <Contact>
  - <Name>
    <value>erratosthenes</value>
  </Name>
  - <TwitterScreenName>
    <value>Erratosthenes</value>
  </TwitterScreenName>
  - <Description>
    <value>Proxymoron: Oxymoron</value>
  </Description>
  - <Location>
    <value>State College PA</value>
  </Location>
  - <TimeZone>
    <value>Eastern Time (US & Canada)</value>
  </TimeZone>
  - <TwitterID>
    <value>web</value>
  </TwitterID>
</Contact>
```

Figure 5.12: Servlet Response for Contact Information Request

So, all information that is centrally stored in the Local Database can be accessed by internal as well as external applications easily. Since the design of Servlets as well as the Database is completely controlled, it is easy to restrict information as well as change the response format of any request to the Servlet. This is also especially helpful if there are rate constraints on the number of requests per unit time to third party databases like that of Twitter. Now that the society structure is clear, a hierarchical Agent society structure can be used for data fusion applications. This is elaborated in the next section.
Higher Layer Design Considerations

Figure 5.13: Higher Layer Design Considerations

This segment uses the architectural structures introduced in previous sections and builds on the theories of higher order Agent Societies and Layered Sensor Characterization. Figure 5.13 presents the main idea which will now be explained using previous designs. As demonstrated, the tasks performed by the lowest layer society can be elaborated as follows:

1. Initialization: As Shown, the algorithm initializes with certain initialization vectors. The selection of these vectors is crucial and has a huge impact on the quality of the sensor array created. For example, if a particular nature of people needs to be made part of the array sensor characteristics, suitable filters need to be
designed. Given that likeminded people share stronger relations, a group of initially trusted users can be the initialization vectors that lead to an array. This was the logic implemented in the Agent Society describes previously. Another example would be the case where a particular geographical area is of immediate interest. Then, the user location tag is the most important initial condition. The algorithm can be supplied by these init conditions via two major ways.

a. Manual: Assigned by a human in the monitoring and control division using human computer interaction (HCI). This has been demonstrated in the Agent society. The initialization parameter vectors of trusted users were manually provided to the algorithm.

b. Dynamic through Feedback: For example, given a certain context, the agent society can use feedback from its understanding of the situation and control these initial conditions for future dynamic sensor arrays.

There can be multiple Agent societies at this lowest level as shown in figure 5.13, each having its own initialization vectors. They pass their outputs to the higher tiered societies (The Controller Agents) that use them to extract information. In this case for example, the lowest level Agents are the ones creating the Sensor arrays of human observers. These Arrays are the “Output Objects” in this case and can be used by higher tier agents to extract real time user information from the blog.

2. Creation of the Dynamic Sensor Array: There can be a number of features of importance for each soft sensor depending on current context. In other words, for every situation being analyzed, there can be a number of desired characteristics in
the I-Space human sensors. A step wise approach of identifying characteristics and looking to see if they match the requirements can greatly help design process. Example: If the Dynamic Sensor Array needs to focus on students on the Pennsylvania State University Park Campus, then there are at least three basic requirements that are immediately obvious. Firstly, that the users need to be students, second that they must be physically be located in State College and third that they be enrolled at Penn State. A modular agent society approach can first create a location based array and then identity students from that and so on. This ultimately results in the desired array with a maximum of sensors qualifying under the conditions specified.

3. Monitoring: Once the output of each agent society is available, it can be visualized using visualization techniques as well as monitored to gauge performance. For example, as soon as a suitable H-Space Array has been created; higher order agent societies can perform the needed analysis on the sensor feeds from them while simultaneously analyzing soft sensor characterization.

Layered Soft Sensor Characterization

As Shown in figure 5.13, a layered sensor characterization model can be chosen. Here is a formal definition of Layered Sensor Characterization (LSC): “LSC for I-Space human sensors can be defined as a Trust-Metadata Characteristic map of individual sensors and the Arrays they are a part of”. In other words, at the smallest granularity, for an agent, a unit source of information is a single human observer who is a part of a larger
group of observers within the Sensor Array. So characterization begins at this level. The Observer will have a multi-dimensional trust value associated with information produced. At the higher second level, the smallest unit is a sensor array or a group of Observers that collectively form a general opinion vector about the context in consideration. So each array, as a single cumulative unit, will have a dynamically changing trust value and its related characterization information like for example a Metadata Tag <User Group: Students>; specifically a Penn State Student Array. So, cumulative trust will be a function of the individual trusts of each user. This goes on up the entire hierarchy of Agents.

The benefit of this approach is the fact that any dynamic array can be gauged on the basis of the cumulative characterization parameters it produces. The target for any Agent society working to create such an array would be to maximize that value.

**On Trust**

Trust, as a quantity, has non-singular dimensionality apart from being subjective as well as asymmetric depending on context. For example, software Agents trying to update the trust values of I-Space human sensors have various considerations to take into account depending on the target scenario. Based on the evidence gleaned from human sensors, each agent society handling information extraction for a particular context will have different trust values for the same sensor. A trained computer scientist for example will have different trust levels within a society looking to gauge terrorist threats than a society looking to create an array of people with a background in education. Also, as
already mentioned, trust is highly variable for human sensors and it is inherently linked to risk. The idea of multidimensional trust in the buyer/seller rating scenario has been described in [34]. Wang et al. calculate the value of trust as the expectation of beta probability density function. From [34], the beta probability density function can be expressed in terms of the gamma function as:

$$\text{beta}(p|\alpha, \beta) = \frac{\text{gamma}(\alpha + \beta)}{\text{gamma}(\alpha) + \text{gamma}(\beta)} p^{(\alpha-1)}(1 - p)^{\beta-1}$$

Where,

$$0 \leq p \leq 1 \text{ and } \alpha, \beta > 0$$

Then the expectation of beta distribution can be calculated as,

$$E(p) = \frac{\alpha}{\alpha + \beta}$$

The authors of [34] then state that the posteriori probabilities of events can also be represented as beta distributions. So if a current binary process (a process that can end up in a total of N+M events of only two types) has N possible ‘e’ event outcomes and M possible ‘f’ event outcomes, then the PDF of event e occurring in future can be expressed as a function of already occurred events or observations by using the conditions:

$$\alpha = N + 1 \text{ and } \beta = M + 1,$$

Where,

$$N, M \geq 0$$

It is concluded that depending on the situation under analysis, the number of different events or outcomes occurring in the future can be more than two which will give a better understanding of the possible outcomes of the process. So for example, in a
multidimensional Trust value estimation scenario for I-Space human observers, there can be many dimensions to the value of trust such as “Knowledge of X”, “Knowledge of Y” and “Knowledge of Z”. In a binary scheme, this can be represented as X and X’, Y and Y’, Z and Z’ where X, Y, Z indicate presence and X’, Y’, Z’ represent absence of the corresponding knowledge. This type of trust characterization based on certain observer traits is critical to gauge their usefulness as sensing resources. All the above can be part of the number of interactions with a particular outcome represented by ‘e’. So,

\[ e \in (X,Y,Z), e' \in (X',Y',Z') \]

So if \( N_e \) represents the number of interactions with these observers with outcome e and \( M_e \) represents the number of interactions without outcome e, then,

\[ \alpha_e = N_e + 1 \text{ and } \beta_e = M_e + 1, \]

And,

\[ \gamma_e = \frac{\alpha_e}{\alpha_e + \beta_e} = \frac{N_e + 1}{\sum_{k \in \{X,Y,Z\}} N_k + 2} \]

Since,

\[ \beta_X = \alpha_Y + \alpha_Z - 1 \]

And so, trust can be found by the normalized values,

\[ P_e = \frac{\gamma_e}{\sum_{k \in \{X,Y,Z\}} \gamma_k + 2}, e \in (X,Y,Z) \]
Chapter 6
A Prototype Cyber Infrastructure Environment: S Space

Design and Implementation

An S-Space Video Sensor System is implemented to survey the various techniques available for object tracking in the domain of image and video content processing. While the final aim is to extract contextual information from behavioral inferences made on the basis of pattern recognition, video and image processing systems also have layers to them in the design sense. For example, detecting a moving object is different from detecting a moving human or animal. Feature extraction and comparison is the additional step required in the latter since mobile objects need to be classified.

Then there is the problem of identifying irregular behavior that might be indicative of negative future occurrences. This needs a deep understanding of the relationships between events where concepts like Bayesian mathematics come into play. In this section the following goals have been achieved:-

1. The first target achieved is the establishment of physical infrastructure needed to facilitate video dataset collection. This is tested by relaying a live video feed to the system that can be used for video processing and information extraction. Client-Server architecture has been used to implement the streaming of live video over the Penn State virtual private network.

2. The second target achieved is the development of object tracking algorithms in MATLAB. Three major tracking algorithms are presented that have different performance benefits in different circumstances.
Physical and Network Infrastructure

Figure 6.1 shows the Client-server model based infrastructure that is used to demonstrate how live video sensor feeds can be multiplexed to the system. An experiment is presented where Apple Mac Book Laptops mounted with Apple iSight web cameras act as mobile video sensor nodes. They are connected to the Penn State virtual private network. An Apple MAC Pro desktop computer is the server that runs the Darwin Media Streaming Server and accepts multiple incoming live video feeds. The client laptop streams live video to the server using the Apple QuickTime broadcasting application.

![Figure 6.1: Video Processing Infrastructure](image)

Hardware and Software used:

1. Video Sensor: Apple iSight; the external iSight's ¼-inch color charge coupled device sensor has 640 × 480-pixel VGA resolutions. It supports video capture at 30 frames per second in 24-bit color with which is ideal for experimentation.
2. Server Platform: Apple Mac Pro desktop, Darwin Streaming Server, QuickTime player (Used to visualize live video feeds at the server.)

3. Mobile Client Platform: Apple Mac Book with the video sensor and QuickTime Broadcaster.

Once the live video is available at the server, MATLAB can be used to demonstrate object tracking Algorithms. The image acquisition and the image processing toolboxes are the primary MATLAB toolboxes that have specialized functions used in these algorithms. The acquisition toolbox can also be used to directly fetch live video from peripheral Video Camera Sensors.

Object Tracking Algorithms

Three major algorithms are surveyed, implemented and discussed in this section. The aim is to provide examples of tracking objects and identifying movement patterns. This is information extraction that lays the foundation to extract contextual information from patterns observed by tracking entities or groups of entities. We begin by the following details of datasets used to demonstrate these algorithms. The algorithms and implementations follow thereafter.

Dataset Details

ETH Video [36] Dataset Details:

(Frame) Width*Height = 720*576 pixels

Video Frame Rate: 25 Frames/second, Data Rate: 504 kbps
IST Video [37] Dataset Details:

(Frame) Image Width*Height = 640*480 pixels

Video Frame Rate: 30 Frames/second, Data Rate: 1248 kbps

INRIA Person Image Dataset [40] Details:

Image Width = 640*480 pixels

Horizontal Resolution: 96 dpi

Vertical Resolution: 96 dpi, Bit Depth: 24 bits

**Covariance Tracking**

Let the observed image be \( I \), and \( X \) be the \( M \times N \times d \) dimensional feature image extracted from \( I \), and then any window that represents the object of interest will have a \( d \) dimensional feature vector set. It is mentioned in [35] that this feature vector can have various attributes like spatial or geometrical attributes as well as others such as color, luminance, and gradient and so on. These features can be associated directly to the pixel coordinates which can help in identification of an object in a sequence of images or frames comprising a video signal

\[
f_k = [x \ y \ I(x,y) \ I_x(x,y) \ ...] \ , \text{ from [35]}
\]

Then an \( M\times N \) rectangular region \( R \) with a \( d \times d \) covariance matrix \( C_R \) of the feature points can be represented as

\[
C_R = \frac{1}{M \ N} \sum_{k=1}^{MN} (f_k - \mu_R)(f_k - \mu_R)^T
\]

Where, \( \mu_R \) is the vector of the means of the corresponding features for the points within the region defined by \( R \).
In this case the diagonal elements of this covariance matrix represent the variance of each feature. Also, the non-diagonal entries represent the respective correlations. So then a covariance matrix of a feature set constrained by a sub-matrix of the entire image matrix combines and extracts relationships dependent on all features mentioned earlier. Coupled with the properties that covariance matrices are scale invariant makes it an ideal detection technique for rapidly approaching or retreating entities that do not maintain shape.

So the first step is to consider an area or region of interest in an image. This could be a non-rigid object that needs to be tracked. The distance between the co-variance matrix for this region and the co-variance matrices of candidate regions that surround the region needs to be minimized to find the precise location of the object in the next frame. So the region with the least distance from the original becomes the next estimated position. However, a simple arithmetic subtraction of the two co-variance matrices cannot measure the corresponding distance since the covariance matrices do not lie in a vector space. So [35] uses a different scheme, in which the metric is defined as:

\[
z(C_i, C_j) = \sqrt{\sum_{k=1}^{d} \ln^2 \beta_k(C_i, C_j)}
\]

Where, \(\beta_k(C_i, C_j)\) are the generalized Eigen values of \(C_i, C_j\), computed by the formula,

\[
\beta_k C_i X_k - C_j X_k, \quad \text{for } k = 1 \text{ to } d
\]

Here, \(X_k\) are the generalized eigenvector. At each frame the region with the smallest distance from the current object model gives the highest probability region of the existence of the object in the current frame. The model can also be updated with a mean
aggregate covariance matrix to count for the changes in spatial geometry of non-rigid objects.

This algorithm has been implemented to perform object tracking.

Algorithm Steps:

1. Detect the “Object” or region of interest within the first frame of the Video Frame Sequence. Detection here means extracting pixel based information that uniquely identifies the entity being targeted for tracking. For the purposes of this Algorithm, location metadata was already available for the entities being tracked.

2. For the First frame, find the covariance matrix for the entity. This forms the base matrix with the covariance matrix reflecting the unique feature set that the entity represents in terms of image information.

3. For all subsequent frames, determine the upper and lower limits of the entity position based on the previous estimated position to delimit the candidate regions in the frame. These upper and lower limits give the bounds for the next candidate region.

4. For the candidate region derived above, construct the covariance matrix and calculate the distance metric from the last derived matrix.

5. From amongst all covariance matrices found for the candidate regions, find the one that has the least distance metric from the covariance matrix of the entity predicted or known in the previous frame.

Observations and Results:

1. Observation Case 1: Single Non-Rigid Object Tracking, ETH dataset [36]
Figure 6.2: Single Non Rigid Object Covariance Tracking (ETH)
2. Observation Case 2 Tracking a non-rigid object, IST Dataset [37]

Figure 6.3: Single Non Rigid Object Covariance Tracking (IST)

3. Observation Case 3: The algorithm is implemented to track two targets simultaneously and to determine their respective movement vectors. This study is
useful in finding the relation between the motion paths of a group of people related or unrelated. Figure 6.4 shows the tracking results.

Figure 6.4: Multiple Non Rigid Object Covariance Tracking (ETH)
Figure 6.5 is plot of the movement vectors of the two tracked entities. In this case, the X and Y axes harbor distance metrics mapped into image pixel value such that the pixel distance can be simply mapped to actual position in the image.

Figure 6.5: Movement Vector Plot for 2 Objects, Covariance Tracking (ETH)

Horizontal distance defines how much the object moved laterally and Vertical distance maps how much an object moved in the plane vertically.

Image Segmentation using Frame Differencing (FD) and Motion History

Frame differencing is a technique where the difference between two chronologically occurring frames of a video sensor feed indicates movement, if the resultant is non zero of course. This technique can also be used to track relatively slow
moving targets. Slow, in this context means observable by the video camera sensor being used. Frame differencing can also be combined with a motion direction detection scheme like finding the Motion History Image or MHI [38]. The approach implemented here thus is divided into two parts. First, the object is detected using frame differencing by the virtue of its motion within the field of observation of the sensors. Then the movement of that detected object is compared between successive frames by analyzing the change in gradients and motion is color coded to indicate findings.

In this implementation the first two consecutive frames from a given sequence of video frames are considered to be the initialization frames. These RGB color frames are converted to gray scale images since color information is of no consequence. Binary Image Thresholding is used to threshold the difference image. Any grayscale image can be converted to a binary format by applying a filter. The filter defines the threshold value for each pixel based on the luminance content in the difference image. Luminance, for each pixel ranges from 0-255 and so a threshold value can be chosen from within that range. If the luminance value of any pixel is lesser than the threshold level, it is assigned the value 0 which indicates the color black and if greater it is assigned 1 which indicates white in Luminance terms.

Then, the consecutive gray scaled images are differentiated and their absolute difference is used to identify the movement between frames. An interesting feature of this system is that the noise collected due to subsequent difference cycles is also filtered by the thresholding process. A smoothing filter can also be applied to do away with excessive noise in the image differences.
A Motion History Image has direction information embedded in it. This can be extracted by taking its gradient [38]. This gradient gives direction vectors that are orthogonal to the boundaries of the moving target entity, through all its phases of motion. What this means is that for locations within the MHI, successive gradients and their respective orientations give an estimate of the direction of motion of the corresponding object. For this implementation,

<table>
<thead>
<tr>
<th>Direction</th>
<th>Color Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right / Up</td>
<td>Yellow</td>
</tr>
<tr>
<td>Right / Down</td>
<td>Pink</td>
</tr>
<tr>
<td>Left / Up</td>
<td>Yellow</td>
</tr>
<tr>
<td>Left / Down</td>
<td>Cyan</td>
</tr>
<tr>
<td>Right</td>
<td>Red</td>
</tr>
<tr>
<td>Left</td>
<td>Green</td>
</tr>
<tr>
<td>Up</td>
<td>Yellow</td>
</tr>
<tr>
<td>Down</td>
<td>Blue</td>
</tr>
</tbody>
</table>

Table 6.1: Color Code for Motion History Movement

Algorithm Variables:

The parameter Threshold (T) provides a handle to the degree of movement expected to be discerned. The value of the variable threshold is directly proportional to the degree of movement to be detected. Smaller changes can be detected for smaller
values of this variable. Filtered images can be used to find the changing gradients that can be used to persist and color code movement. A decay factor ($\alpha$) is used to control the movement decay and within each iterative cycle. $\alpha$ is inversely proportional to the amount of motion history information desired for the data set as documented in the following plots.

Observations and Results

1. Observation Case 1: Dataset [37]

Figure 6.6: Frame Differencing with $\alpha = 250$ and $T = 80$ (IST)

Figure 6.6 shows a differenced frame, its motion history as well as the color coded motion history. The decay factor alpha has a value 250 and threshold is 80 to filter out minor movements.
Figure 6.7 shows the same image data set with the decay factor of 50 with a threshold of 80.

Figure 6.7: Frame Differencing with $\alpha = 50$ and $T = 80$ (IST)

Figure 6.8 shows a differenced frame, its motion history as well as the color coded motion history with decay factor 250 and threshold 20. As observed, due to the low threshold value, minor movements are also successfully detected that were being filtered out in the higher threshold case. Four frames from the detected sequence of the moving object are shown in Figure 6.9.
Figure 6.8: Frame Differencing with $\alpha = 250$ and $T = 20$ (IST)

Figure 6.9: Frame Differencing Detected Sequence $\alpha = 250$, $T = 80$ (IST)
2. Observation Case 2: Dataset [36]

Tracking Multiple Entities: $\alpha = 50$ and $T = 100$.

Figure 6.10: Frame Diff. Multiple Object Detection $\alpha=50$, $T=80$ (ETH).
Weaknesses: A few weaknesses of the algorithm were recorded as evidenced in the following frame. There are instances of detection window fusion for multiple mobile entities. A fix is to implement Kalman Filtering to distinguish between individual entities. Partial or multiple detection can also be seen from this frame.

![Image of multiple object detection weaknesses](image)

**Figure 6.11: Frame Diff. Multiple Object Detection Weaknesses (ETH)**

### The Histogram of Oriented Gradient (HOG) Algorithm

Histograms of Oriented Gradients or HOG descriptors are a type of feature sets that can be used to detect specific objects on the basis of pixel gradient analysis [39]. The primary idea is to match gradient orientations for localized Image regions. The original image is divided into smaller units called “cells”. A histogram of gradients is found for each cell based on the edge orientations of the pixels within the cell. All these histograms, contribute to a weight that forms the final descriptor for the image.
Additional normalization can be performed across multiple cells which introduces more resilience towards changes in shadowing or illumination. Here are the steps of the implemented algorithm. A list of 9 different images is used as the test images; 5 of which are known to be of humans and 4 are non-human. 4 of the 5 person images are from the INRIA [40] dataset. Each of these is tested against a repository of 80 INRIA images.

1. Divide the test image and reference image into cells. Follow steps 2 through 4 for both images.

2. For each cell find the gradients in the X and Y directions. Calculate the corresponding gradient magnitude:

   \[ |G| = \sqrt{l_x^2 + l_y^2} \]

   Where, \( l_x \) and \( l_y \) are the x and y derivatives of each cell within the image which result from the convolution of the cell matrices with filter kernels.

   \( D_x = [-1 0 1] \quad \text{And,} \quad D_y = [-1 0 1]^T \)

3. Calculate the gradient orientation \( \phi \) by,

   \( \phi = \arctan\left(\frac{l_y}{l_x}\right) \)

4. Create the cell histograms for each cell. The histogram channels can be distributed over 0 to 180 degrees or 0 to 360 degrees. N channels can be used. These two conditions form the input variables in the algorithm and have a direct impact on the results. Based on the findings in [39], 9 channels and unsigned gradients (0 to 180 distributions) are used for this algorithm.
5. Use a similarity measure like Euclidean metric or Cosine similarity to find the distance between the histogram vectors for the two images.

\[
\cos(\theta) = \left( \frac{x_1 y_1 + x_2 y_2 + \ldots + x_n y_n}{\sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} \sqrt{y_1^2 + y_2^2 + \ldots + y_n^2}} \right)
\]

Where,

\[ X = (x_1, x_2, \ldots, x_n) \]

And,

\[ Y = (y_1, y_2, \ldots, y_n) \]

Are the two vectors

6. Based on a threshold, decide whether a positive or negative match was detected.

7. For this algorithm, normalization was not performed and the threshold was taken to be 0.50

Observations:

Person Test Images

Figure 6.12: Histogram of Oriented Gradients Person Test Images (INRIA)

Non Person Test Images
The Following Figures (6.14 - 6.22) show the image and their corresponding HOG based similarity measure (probability of the image containing the human form) as measured against 80 person images from the INRIA Persons dataset. The threshold value of 50% works quite well for the dataset in consideration.

Correct Non-Detection: 

Result: N
Correct Non-Detection:

Figure 6.15: Histogram of Oriented Gradients Non-Person image b Result

Correct Non-Detection:

Figure 6.16: Histogram of Oriented Gradients Non-Person image c Result
Correct Non-Detection:

![Image of non-detection result](image1)

**Result**: N

Figure 6.17: Histogram of Oriented Gradients Non-Person image d

Correct Detection:

![Image of detection result](image2)

**Result**: Y

Figure 6.18: Histogram of Oriented Gradients Person image a
Correct Detection:

Figure 6.19: Histogram of Oriented Gradients Person image b Result

Correct Detection:

Figure 6.20: Histogram of Oriented Gradients Person image c Result
Correct Detection:

Figure 6.21: Histogram of Oriented Gradients Person image d Result

False non-detection with person image number 5:

Figure 6.22: Histogram of Oriented Gradients Person image e Result
Results are shown as “N” for a correct non human detection and “Y” for a correct human detection, and “N” for a failure to detect a human. It is seen that 8 out of the 9 images were correctly detected to be either human or non-human with a similarity threshold of 0.50 as already mentioned.

Improvements can be made to this algorithm to have better results. The threshold value can be increased for a more robust match by implementing the normalization step which was not used in the algorithm. Also, for this experiment, no prior metadata about the probable location of the human within the image was used to train a model. A direct HOG descriptor comparison was used on the entire image which also takes other objects into consideration that lowers the threshold value.
Chapter 7

Wireless Ad-Hoc Mesh Network Formation over Bluetooth*

*Referenced from Research conducted at the Mine Safety Appliance Company, Cranberry Twp, PA.

Bluetooth Basics

The Bluetooth wireless technology is a short-range communications technology intended to replace the cables connecting portable and/or fixed devices while maintaining high levels of security. The key features of Bluetooth technology are robustness and low cost. Low Power is generally stated as a positive aspect of Bluetooth also, however, that is a relative term and cannot be mentioned in general terms. For many applications though, it is quite suitable in terms of its low power consumption. The Bluetooth Specification defines a uniform structure for a wide range of devices to connect and communicate with each other. The structure and the global acceptance of the technology mean that any Bluetooth enabled device, can connect to other Bluetooth enabled devices located within its range [41].

A critical design issue is accuracy and reliability of the data being transmitted. In Bluetooth, the Automatic Repeat Request or ARQ is used at the link layer which accounts for data reliability. TCP can also be used over Bluetooth. In that case, the connections enjoy all of the benefits of reliability and congestion control that TCP provides. Any of the TCP modification or enhancement for wireless networks can be used to further boost throughput.
Connections between Bluetooth enabled electronic devices allow these devices to communicate wirelessly through short-range, ad hoc networks known as Piconets. Piconets are established dynamically and automatically as Bluetooth enabled devices enter and leave radio proximity meaning that the user or sensor system can easily connect whenever and wherever convenient.

Each device in a piconet can also simultaneously communicate with multiple Bluetooth devices within that single piconet and each device can also belong to several Piconets simultaneously. In such a scenario, the network is called a Scatternet and the device that exists in both networks does a Time Division Multiple Access

A fundamental strength of Bluetooth wireless technology is the ability to simultaneously handle different transmission data types. This means that data as well as voice transmissions are parallel supported. Besides, ease of deployment and flexibility also make Bluetooth a suitable technology for wireless sensor ad-hoc networks. The only condition being that the sensors should be conditioned to be deployed within physical range described in the Bluetooth Specification.

The range of Bluetooth technology is application specific. The core specification mandates a minimum range of 10 meters or 30 feet, but there is no set limit and manufacturers can tune their implementations to provide the range needed to support the use cases for their solutions. So depending on the S-Space implementation, Bluetooth devices can be interfaced with different sensors and tuned to the necessary range. Table 7.1 shows the effective range of the Bluetooth [41]:

Table 7.1: Effective Range of Bluetooth
<table>
<thead>
<tr>
<th>Class</th>
<th>Maximum Permitted Power (mW)</th>
<th>RANGE (Approximate, meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>100</td>
<td>Up to 100</td>
</tr>
<tr>
<td>Class 2</td>
<td>2.5</td>
<td>Up to 10</td>
</tr>
<tr>
<td>Class 3</td>
<td>1</td>
<td>Up to 1</td>
</tr>
</tbody>
</table>

Table 7.1: Bluetooth Class based Range

**Bluetooth Operation Concepts**

It is useful to begin this discussion with the Bluetooth Protocol Stack. Parallels can be drawn between the protocol stack and the Open System Interconnections (OSI) Model. Figure 7.1 shows the Bluetooth Stack.

Applications use the RFCOMM serial interface layer to create connections. One of the main tasks of this layer is to provide a standard conversion format for any variable formatting data received from applications. In other words, any data received from the application layer is formatted to a “service structure” or units of data recognizable by the lower layers.

The L2CAP layer multiplexes any data from the higher layers and converts it to the required format.

The Host Controller Interface Layer is the interface between the Software and the Hardware Segments in the Bluetooth Specification.
The Link Manager manages links to various Hosts and is similar in functionality to the OSI Transport layer.

The Link Controller controls physical links as well as the frequency hopping sequence and is akin to the OSI Network Layer.

The Baseband acts as the Data Link Layer and the radio is the physical layer that modulates the Baseband signal. More functionality of these Layers will be identified later in this chapter as needed.

![Diagram of the Bluetooth Protocol Stack]

Figure 7.1: The Bluetooth Protocol Stack
Piconet Master and Slaves: The device in a Piconet that provides the clock synchronization reference and Hopping Sequence reference is known as the master. All other devices are known as slaves since they are synchronized to the clock and the hopping pattern of the master so that they can communicate [41]. As mentioned, a group of such devices synchronized to this common clock together with the master device form a Piconet.

Radio and Radio Channels: The Bluetooth equivalent of the physical layer and operates in the industrial, scientific and medical (ISM) Radio band at 2.4GHz. The technology has a bit rate of 1 Megabit per second (Mbps) which can be extended to an Enhanced Data Rate of 2 Mbps to 3 Mbps [41]. The technology implements frequency hopping and during operation, a physical radio channel or frequency is shared by a group of devices that are synchronized to a common clock and the frequency hopping pattern.

Frequency Hopping: Devices in a Piconet use a specific frequency hopping pattern which is algorithmically determined by the Bluetooth specification address and master clock. The basic hopping pattern is a pseudo-random pattern of the 79 frequencies in the ISM band [41].

Adaptive Frequency Hopping: The frequency hopping pattern can be adapted to exclude a portion of the frequencies that are used by interfering devices so that co-existence is improved [41].

Time Slots and Packets, Full Duplex Transmission: In the Bluetooth Specification, the frequency channels are further divided in TDMA like time slots. Data
packets are transmitted in each of these slots. Based on the piconet traffic patterns and implementation, a number of consecutive slots can also be allocated to a single large packet [41].

Control Layers: Bluetooth uses a layered link and associated control protocol type model. The hierarchy of channels and links above the physical channel is physical link, logical transport, and logical link and logical link control and adaptation protocol (L2CAP) channel [41].

Physical and Logical Links: Within a physical channel, a physical link is formed between any two devices that transmit packets in either direction between them. This physical link is used as a medium for one or multiple logical links [39]. These logical link profiles support communication patterns like unicast synchronous, asynchronous, isochronous, and broadcast traffic. Traffic multiplexing is implemented to multiplex logical data onto the physical link [41].

Link Manager Protocol (LMP): Other than the facts already mentioned, it also needs to be stated that the link manager protocol (LMP) is a control protocol for the baseband and physical layers. It is implemented over logical links in addition to user data [41].

L2CAP: As mentioned, the L2CAP layer provides a channel-based abstraction to applications and services. It carries out segmentation and reassembly of application data and multiplexing and de-multiplexing of multiple channels over a shared logical link [41].
Bluetooth Terminology:

1. Pairing: Refers to the initial authentication users before the connection is established. A key is exchanged to determine whether authentication exists.
2. Fixed Pin Exchange: For pairing a fixed pin or key is set that is verified by each node before successful connection establishment.
3. Variable Pin Exchange: A dynamic pin is used by one device which is specific to this connection and the pin or key is expected to be input at the other device.

National Semiconductor LMX9838 Bluetooth Modules

The LMX9838 [42] is a National Semiconductor Bluetooth Serial Port Module. It is a class 2 Device with the complete Bluetooth 2.0 Stack with the Baseband and Link Manager. It supports protocols such as the L2CAP, RFCOMM and Service Discovery Protocol (used by Applications to receive information about services available at local as well as other remote Bluetooth devices.)

Due to its small size and 2.4 GHz radio it is a good choice for developing ad-hoc networks and subsequent routing protocols. Furthermore, Bluetooth profiles like the Generic Access Profile (GAP), the Service Discovery Application Profile (SDAP), and the Serial Port Profile (SPP) are also supported. Due to the presence of a configurable Service Database (SD) to fulfill additional profile service requests on the host, it supports point-to-point and point-to-multipoint link management supporting RFCOMM data rates of up to 704 kbps.
Bluetooth Sensor Network Creation and Routing Algorithms

The Bluetooth standard can be used to develop wireless mesh networks in the field and on the fly. This is demonstrated in this section. To experiment with network topologies and to test routing packets from one Slave Bluetooth device to another within the network, two different experimental procedures are performed.

1. Procedure 1: Bluetooth Ad-Hoc Piconet Development Procedure. Using National Semiconductor LMX9838 Bluetooth evaluation Units [42] to create a Piconet with the Serial Port Profile (SPP) to exchange data. The nodes choose “friendly” devices based upon an assigned Identity value which is called the devices’ “Friendly Name”. For example, in this case, all known devices have been assigned a Group ID: A. that is verified before pairing is initialized with any remote device. It needs to be noted that this parameter is different from the “Pairing Key” that is exchanged between devices to create the Bluetooth connection. The Key also known as a PIN code can be any 16 Byte UTF-8 (8 Bit Universal Character Set/Unicode Transformation Format) string or 4 Byte Fixed Pin depending on device complexity. In this case, the LMX9838 are Limited Input Devices with 4 Byte PIN exchanges. So, as a design goal, only devices that have the friendly group name “A” will be paired with using the Fixed Key Exchange method.

2. Procedure 2: The second procedure deals with verifying the Piconet created in Procedure 1. The goal is to create a Data Link Layer Routing Protocol to route data between slave devices in the Piconet. It Link Layer since it uses the 48 Byte
unique Bluetooth Address to route information between slave devices. The Algorithm is implemented by the Master device so that it is capable of routing packets between slaves that do not share a physical link in the star topology.

**Bluetooth Ad-Hoc Piconet Development Procedure**

Aim: To Develop a Test bed for the National Semiconductor LMX9838 Bluetooth Modules and show proof of concept that they can be used to create an ad-hoc wireless sensor to interface sensors which can be used to sense the environment.

![Bluetooth Piconet Diagram](image)

*Figure 7.2: A Bluetooth Piconet using the SPP Bluetooth Profile*
Procedure 1 Setup Conditions: Specifically, a test bed is constructed, in which, within a region there are \( N \) active devices \((N \geq 2)\). One device is assumed to be the Master of the future ad-hoc network and this Bluetooth device is connected to a Host Controller Microprocessor as shown in Figure 7.3. The Host Controller has the Middleware Application that creates the network.

![Host Controller - LMX9838 Connection Schematic](image)

Figure 7.3: Host – LMX9838 Connection Schematic

The Host Application issues specific commands to the LMX9838 Master and uses it to create a Bluetooth ad-hoc network with only those devices that are recognized to have a particular Group Identity Value.

1. Initially, \( N = 5 \) out of which, 3 Bluetooth Devices are friendly and one becomes the master in a star network topography.

2. The Master is connected to a microprocessor that initializes pairing requests.

3. The Slaves receive and accept requests on the basis of the “Fixed Pin Exchange” pairing procedure. The default pin used is 0000
4. The Bluetooth security mode used for LMX9838 is “mode2” [43]. The Master already has the knowledge of the “Bluetooth Friendly Names” of the two nodes it wants to pair and connect with. Other Bluetooth devices in the area have different names.

**Algorithm Details:** An algorithm is presented that runs on an i386 microprocessor directly connected to the master Bluetooth device via the Serial Port RS-232 9 Pin Connector.

1. **GIAC (General Inquiry):** Create the GIAC command string and send it via serial port to the master. The master then starts polling (issuing the GIAC inquiry) and waits for the response from slave listeners. Replies comprise the respective slave device Bluetooth addresses.

2. After all active slave devices reply, check the status of the master. If this is verified to be okay, proceed. Else waits for a while and reinitialize algorithm sequence from step 1.

3. **Friendly Name Request:** Create and send the friendly name request command to all the devices that responded, by using their respective Bluetooth addresses received as a result of the general inquiry. Check status and proceed if okay. Else wait and retry step 3. After N tries reset device and retry step 3.

4. **Device Name Authentication:** From the list of received Friendly names, check for recognized device names. Proceed to the following pairing procedure only for devices with authenticated device names. Check status and proceed if okay. Else wait and retry this step. After N tries reset device and repeat.
5. Service Discovery Application Profile (SDAP) Connect: Create and issue the SDAP Connect Command. SDAP describes how a Bluetooth Device should use Service Discovery Protocol (SDP) to discover services on any remote Bluetooth device. It illustrates several approaches to managing the device discovery via Inquiry and Inquiry Scan and service discovery via SDP. Check status and proceed if okay. Else wait and retry this step. After N tries reset device and restart step.

6. SDAP Service Browse Serial Port Profile (SPP): Create and send this command to the master. The master device issues this command to fetch the available radio frequency communication (RFCOMM) layer ports and the respective services available at the remote recognized Bluetooth devices. The open port numbers are needed to connect to the remote device. Check status and proceed if okay. Else wait and retry this step. After N tries reset device and restart step.

7. SDAP Disconnect: Create and send the command string to the master. There can be only one SDAP connection at a time and this will be needed at a later stage for repeating similar requests. Check status and proceed if okay. Else wait and retry this step. After N tries reset device and restart step.

8. Establish SPP Connection: Create the command string and send it to the master device. This command is issued to connect to the remote device using the Bluetooth SPP profile. It is created using three parameters. First, the Bluetooth address of the remote recognized devices (via GIAC). Second, the local master device RFCOM port for the SPP connection and lastly, the remote RFCOM open port (via SDAP).
9. Check status: Check status of the system. If stable then all the recognized devices have been connected to and the PAN has been formed successfully.

10. Verify the sensor network by using two techniques. One is to check whether the expected network was formed by visual inspection of the status LEDs on each device. For the LMX9838 devices, a steady Blue status LED indicates a connected device. Second is to create a routing algorithm to transmit packets of information from one slave device to the other. Since a direct link between them is absent, it is safe to assume that any successful routing of packets between two slaves will indicate the successful creation of a SPP Bluetooth link between them all.

**Observations:** The success of this algorithm was established by the visual inspection of the status LEDs on each of the friendly slaves and the master. The observations of the second confirmatory test are documented under objective 2. This was tested by the virtue of two facts:-

1. The Connect LEDs on the Bluetooth devices showed that only the friendly devices were successfully paired and connected to the master device.
2. A routing implementation designed to route packets of information from one slave to the other worked perfectly. This is documented ahead.

The following section describes Procedure 2 in detail.
Routing Protocol Design Details

Procedure 2: Show proof of concept that even though no physical link exists between any two slaves, the master can act as a router and forward packets of information between the two slave devices. In other words, a logical link exists between slave devices. Also, confirm that the SPP network was successfully created by verifying data transmission.

1. Assumption: The sender slave (The slave device that needs to route a packet via the master) knows the Bluetooth address of the destination slave (the slave device that will receive the data)

2. A data link layer communication protocol is presented. The master parses the packet payload sent to it by the sender slave. Figure 7.3 shows the packet byte wise format defined by the National Semiconductor LMX9838 specification [44]:

| SD | Header | Payload | ED |

Figure 7.4: LMX9838 SPP Packet Format

SD: Packet Start Delimiter, Length: 1 Byte

Header: Packet Header, Length: 5 Bytes

Payload: Frame Data, Length: X Bytes (Max 333 Data Bytes per packet)

ED: Packet End Delimiter, Length: 1 Byte
The header in the packet comprises of 4 sub-fields the first one being the Packet ID. The packet Type ID field is 1 byte long and identifies the nature of request being sent. It can have values that represent one of the four possibilities, Request, Response, Indicate and Confirm. The second sub-field within the header is the opcode which is also 1 byte in length and it identifies the actual command within the packet. For example, for the Serial Port Profile, a request with incoming data can be represented by the byte represented by the two nibbles, 0x10. A complete list of these values can be found in [45]. The third sub-field in the header is the 2 byte representation of the length of the data being included in the payload field of the packet. Like mentioned the maximum value of this length can be 333 data bytes per packet for the LMX9838. The final field is the 1 byte Checksum which is a block check character checksum of the bytes contained in the Packet type, Opcode and Data Length fields/sub-fields. This is calculated as the lower byte of the sum of all bytes present in the abovementioned fields in the packet.

3. While the design of the packet being sent is dependent on the standard implemented for the LMX9838, the payload can be designed depending on particular usage.

   It has been constructed to have the following format that works well for this scenario while being very simple in design:
Figure 7.5: Payload Format for Data Link Layer Routing

Data: 1 byte data sent by sender slave to the receiver slave device

Destination Bluetooth Address: 48 Byte Unique Bluetooth Device Address

Algorithm Details

1. Sender slave sends the routing / testing frame to the master
2. The master parses the payload and extracts the data and destination address. If it does not recognize the frame, it does not act on it.
3. The master creates another frame with the data and finds the local RFCOM port corresponding to the destination address received from the sender
4. The master then routes the packet to the destination slave.
5. The destination slave sends an acknowledgement that is read by the master and the process is complete.

At the master, the Bluetooth address is parsed and verified and mapped to a local RFCOMM port. This unique mapping between the local RFCOMM ports of the master and the remote device addresses of the slaves is accomplished during the SPP connection phase). Thus any data that the master sends to a local RFCOMM port is routed through to the corresponding device. No further information about the Bluetooth address is required. This means that even at the slave devices, only the local port bound to the master is
required to send data. This is a feature of the Serial Port Profile where the Bluetooth connection acts as a serial port cable replacement strategy.

**Observations:** The data is successfully transmitted and the network has been tested by using application layer terminal emulation programs (Example: HyperTerminal on Windows and Minicom / Cutecom on Linux) at the sender as well as the receiver slave. The sender slave is connected via the serial port to a computer running the Windows XP operating system and the receiver slave device is connected to a computer with the Linux operating system. Both machines have the application layer terminal emulation programs that can represent the transmitted data bytes as string or hexadecimal values.

**Conclusion**

It is successfully demonstrated that Bluetooth enabled sensor nodes can be easily used to form ad-hoc sensor networks once deployed in the field. The security features provided by Bluetooth can be leveraged on, to protect the network from external attacks. Especially with the Bluetooth system implementing frequency hopping, it would be difficult for an external agent to snoop on the sensor feed. Besides, additional logic can be implemented in the middleware to recognize friendly devices and only those devices could be paired with. However, this does raise the question that if friendly names were compromised then non-friendly nodes could join the network posing as friendly nodes. This can be solved by implementing a periodic refresh of the friendly name as that is programmable in most Bluetooth devices. Keeping track of such a periodic refresh would be very difficult for non-friendly nodes.
Chapter 8
Application Scenarios

A fusion system developed from the infrastructure discussed in this report can be used to assist in many real world circumstances and applications. One such implementation is to evaluate extreme events. Figure 8.1 shows the scenario depicting the general overview of a typical problem where the above methods can be effectively used.

Figure 8.1: Extreme Event Prediction: General Idea
The Application can be described using the following definitions:-

1. Extreme Event: An Extreme Event refers to any event that in any way harms or disrupts human life or systems or related dependencies and dependents. Examples range from terrorist attacks to earthquakes to robberies. There is a threat level associated with each extreme event which is a function of the intrinsic danger that the event poses and the real time unique response to that event from the entities in its surroundings.

2. System: The “system” here refers to the entire data fusion model that houses the intelligent Agents and other techniques as well that can help identify the threat levels in any situation.

3. Pointer Event: *Pointer events* are smaller events that may indicate the future possibility of an Extreme Event. For example, in case of a terrorist attack, any smaller event that seems independent but is actually a part of the larger attack can be thought of as a “pointer” i.e. an event that might lead to a major disaster. Another example would be that of a leaking gas valve that could potentially lead to a major fire in the neighborhood. In that case, the leaking gas valve would be a pointer event pointing towards the possibility of fire. This can be tied to the concept of Bayesian Network Events that have a certain probability with which they result in future events.

   In Figure 8.1, the x axis holds the pointers and the y axis shows time. The smaller pointers point to an extreme event as time progresses with a certain probability. The Probability of the Extreme Event occurring depends on the individual probabilities of these pointer events. So at the “present” time stamp a
system that senses these pointer events to try and judge whether they point to an extreme event in future or not is a critical resource to have. Such a prediction system that can be trained then to avert major disasters would be a strong disaster management resource. A data fusion system utilizing the help of a huge population of human observers as well as S-Space traditional hard sensors that can extract patterns from data and events spanning the human landscape is needed and this report is a step towards that.

The idea of a hierarchical design structure for such a system utilizing intelligent agent societies is shown in Figure 8.2. The complex task of characterizing a chain of live events can be broken down into simpler tasks by using a hierarchical model for the intelligent agent system. The lowest tier (tier1) agents handle individual sensor streams and report their findings to the higher tier which then tries to find the co-relation between these findings and then passes a poll based decision to the next higher level agents.

All agents have unique capabilities at all levels. The highest level agents will have the capability to co-relate all possibilities and see the big picture and predict how the events will pan out in future. Logically, as we go higher up in the agent hierarchy, knowledge about the entire event increases. This is due to the fact that the higher tier agents can see the “patterns” in data observed and reported by lower tier agent societies.

Another possible application of such a system would be to identify and analyze data from observers involved as participatory sensing nodes. Pre-formatted text can be analyzed and parsed easily by agent based systems.
The point made in Figure 8.2 is that pattern recognition at the higher tier agents will yield more information about the events as a whole and can thus predict on the basis of its evaluation. This was the intent when it was mentioned previously that higher tier agents can see patterns in wide spread phenomena that are difficult to analyze by locally distributed sensors. This situation is of particular interest in the case of analyzing the human landscape.

Figure 8.2: Agent Hierarchy
Chapter 9

Summary and Recommendations

Limitations and Future Research

The aim of this research was to develop an infrastructure model and show experiments proving design feasibility. This system demonstration uses Twitter as its I-Space Human Sensor Platform. There are some issues with this design then. One is that it is a third party application and neither the NC2IF nor Penn State has any direct control over what that application decides to support. There needs to be a solid replacement of Twitter that can then be used as the platform to develop these real time arrays of human sensors.

Besides that, Twitter is meant primarily for commercial entertainment purposes. The application that replaces it needs to be more specific in terms of design since it will finally facilitate actual information extraction from human reports. However, a more realistic design step can be to implement a “Message Structure” for trained and trusted users for use over Twitter. This structure would be easier to parse and extract information out of. So, in other words, a portal for information submission from the general user population needs to be set up. This will better support soft sensor characterization as well as the hierarchical characterization introduced in this work.

As far as image processing is concerned, MATLAB has the distinct disadvantage of being slower in operation than programming languages like C and C++. As future
research, all algorithms need to be ported to either of those languages to make the system more realistic and practical in real world implementation terms.

**Experiment Design Requirements**

Elaborate experiments need to be designed in future to study data fusion from multiple sources whether they exit in the I-Space or S-Space or H-Space. Apart from the experiments already mentioned, the “analyzing and characterizing the human sensor problem” needs to be tackled. A framework also needs to be designed or decided upon that can reliably reflect the values of trust as well as observer information from I or H Space. Also, this information should lie in a domain that needs to be easily visualized for human in the loop monitoring as well as easily interfaced with programs and routines that require strong formatting and less contextual information to be better suited to machine learning. Further, the characterization should be robust as well as secure for real world applications.

**Result Analysis and Conclusion**

1. **I-Space Agent Society**

   The agent society was used on the premise that such a distributed system can greatly simplify the design process as well as provide flexibility and modularity in design. This was clearly achieved and demonstrated. The I-Space human observer domain can be completely analyzed using Cougaar as the agent
architecture. Further, language processing also can be handled easily by agents. Different frameworks can be integrated into these agent hierarchies. An addendum to the flexibility of this system arises from the capabilities of Cougaar based agents in being flexible in the sense that different functionalities can be added and removed dynamically to agents using plugins. Also, a hierarchical agent structure can ease the task of data analysis.

2. S-Space Image Processing

Infrastructure: The Client-server model based physical infrastructure can be implemented to act as a backbone for the image processing video sensor feeds as shown in this report. Different Client-server pairs can be chosen depending on the operating system being used at the server and client.

Tracking Algorithms: Covariance tracking is a strong tracking technique that can be used along with frame differencing for motion detection as well as HOG for human shape detection.


Bluetooth devices can be used to create ad-hoc wireless networks and can be used to transmit information that can originate at sensors interfaced with those devices.
References


[30] “Natural Languages and the Chomsky Hierarchy”; A Kornai; Proceedings of the second conference on European chapter of the Association for Computational Linguistics; Geneva, Switzerland; Pages 1 – 7; 1985

[31] “Quantcast”; http://www.quantcast.com/twitter.com


[36] ETH Walking Pedestrian Dataset (EWAP);
http://www.cs.ucf.edu/courses/cap6412/fall2009/misc/ewap_dataset_full.tgz

[37] The Pennsylvania State University College of IST, NC2IF Pedestrian Video Dataset.


[40] “Histograms of oriented gradients for human detection”; Dalal, N. Triggs, B

http://www.bluetooth.com/English/Technology/Works/Pages/default.aspx

[42] “National Semiconductor LMX9838 Overview”;
http://www.national.com/mpf/LM/LMX9838.html#Overview


[44] “National Semiconductor LMX9838 Datasheet”;


Appendix A

Twitter Terminology

This is a list of some Twitter related terms that might be needed to understand some Twitter Specific Language used in this report.

Tweet

Each Twitter post with a maximum length of 140 characters is known as a tweet. These are also referred to as “updates”. For our context, this is the Real Time Human Sensor Feed.

Timeline

A User’s Timeline refers to a chronological sequence of their tweets. The Timeline in the most general terms is a chronological sequence of tweets.

Follower

Twitter has the concept of one-way relationships and two way relationships. In Twitter terms, to "follow someone" means to subscribe to their tweets. This means that the followed User’s tweets appear in the follower’s update section for them to read. It is not necessary that the followed user reciprocates by following the follower in turn thereby indicating a unidirectional relationship.

Friendship

This report uses the term friendship for the two way relationship where two users follow each other. Generally, that is a stronger indication of a relationship than a one-way relationship of follower/followed.
Reply

A Reply is when a user responds to a tweet from any other user. These are completely public in nature and appear as public information.

Direct Message

A direct message from a user to the other is like a personal and private message. It is akin to a limited length textual inbox message that only the recipient can view. A user can only send a direct message to another user that is following them.

Favorites

If users like other users’ tweet, they can flag the tweet to be one of their favorites. This generally indicates their favorable opinion of whatever has been stated or implied in the tweet.
Appendix B

Database Schema

The following table shows the database schema used to implement the back end of all data fetched from Twitter via the Twitter API.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CHAR</td>
</tr>
<tr>
<td>StatusID</td>
<td>MEDIUMTEXT</td>
</tr>
<tr>
<td>TimelineID</td>
<td>INT</td>
</tr>
<tr>
<td>ID</td>
<td>BIGINT (Primary Key, Not Null)</td>
</tr>
<tr>
<td>Favourited</td>
<td>TINYINT</td>
</tr>
<tr>
<td>USERID</td>
<td>INT (Primary Key, Not Null)</td>
</tr>
</tbody>
</table>
Table: *TwitterFavorites*

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>ID</td>
<td>INT</td>
</tr>
<tr>
<td>Text</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>UserFromID</td>
<td>INT</td>
</tr>
</tbody>
</table>

Table: *TwitterFollowedUsers*

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>Screen_Name</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>ID</td>
<td>INT (Primary Key, Not Null)</td>
</tr>
</tbody>
</table>
### Table: *TwitterMessages*

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>Message</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>Timestamp</td>
<td>DATETIME</td>
</tr>
<tr>
<td>ID</td>
<td>INT</td>
</tr>
</tbody>
</table>

### Table: *TwitterUserLocation*

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>INT (Primary Key, Not Null)</td>
</tr>
<tr>
<td>Location</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>Timezone</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>Column Name</td>
<td>Column Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Name</td>
<td>CHAR</td>
</tr>
<tr>
<td>Screen_Name</td>
<td>CHAR</td>
</tr>
<tr>
<td>ID</td>
<td>INT (Primary Key, Not Null)</td>
</tr>
</tbody>
</table>

Table: *UserTalkRepository*

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserFromId</td>
<td>CHAR (Primary Key, Not Null)</td>
</tr>
<tr>
<td>UserToId</td>
<td>CHAR (Primary Key, Not Null)</td>
</tr>
<tr>
<td>Count</td>
<td>(Default 0)</td>
</tr>
<tr>
<td>BfollowsA</td>
<td>INT</td>
</tr>
<tr>
<td>Favorites</td>
<td>(Not Null, Default 0)</td>
</tr>
</tbody>
</table>